الجمهورية الجزائرية الديمقراطية الشعبية
REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

وزارة التعليم و البحث العلمي
MINISTERE DE L'ENSEIGNEMENT ET DE LA RECHERCHE SCIENTIFIQUE

ECOLE NATIONALE POLYTECHNIQUE

DEPARTEMENT : ELECTRONIQUE



# PROJET DE FIN D'ETUDES

SUJET

## CONCEPTION D'UN SYSTEME MINIMAL A BASE DU MPU 6809

## _ PARTIE SOFTWARE _

Proposé Par : Mme HAMAMI Etudié par : M. FODIL
M. LAMMALI
Dirigé par : Mme HAMAMI

PROMOTION : JANVIER 1987

E.N.P. 10, Avenue Hacen Badi – EL-HARRACH – ALGER

# DEPARTEMENT D'ELECTRONIQUE



## THEME

# * MONITEUR D'UN SYSTEME MINIMAL *
# A BASE DE :

M.FODIL
M.LAMMALI

PROMOTION JANVIER 1987

# REMERCIEMENTS

Comme toujours, la réalisation d'un projet de fin d'études implique de nombreuses personnes:

Nous tenons d'abord à remercier notre encadreur, Mme HAMAMI, pour son aide précieuse et son suivi.

Nous remercions également l'ensemble du personnel du laboratoire de recherche du C.E.R.I, pour nous avoir accueillis et nous avoir facilité l'utilisation et l'exploitation du matériel du laboratoire.

Et que tous ceux qui ont contribué à la réalisation de ce mémoire, trouvent dans ces lignes l'expression de notre profonde gratitude.

# TABLE de MATIERES

# I- INTRODUCTION

Un micro-ordinateur d'évaluation est un systeme minimal, de faible cout, destiné à évaluer les capacités et possibilitées d'une famille de microprocesseur. C'est un moyen trés pratique pour enseigner les techniques de programmation en assembleur et développer des applications particuliéres à l'utilisateur.

Le systéme d'exploitation, qui contient les programmes gérant le systéme, agit comme intermédiaire entre l'utilisateur et le matériel. Il constitue en quelque sorte l'âme du systéme.

Pour une carte d'évaluation dépourvue de mémoire de masse, ne contenant que le microprocesseur, quelques interfaces d'entrées-sorties, et une mémoire de faible capacité, le systéme d'exploitation se réduit à sa forme la plus simple : le moniteur.

Le moniteur contrôle toutes les fonctions dites de bas niveau; comme la réception des touches frappées au clavier, de leur affichage, et l'initialisation correcte des circuits périphériques.
Le moniteur occupe un éspace mémoire de quelques kilo-octets, et s'intégre, par conséquent, aisément dans une mémoire morte.

Notre étude, qui aura pour but la réalisation d'un moniteur pour un systéme minimal d'évaluation à base du microprocesseur Motorola 6809, comprendra trois grandes parties:

Tout d'abord, nous débuterons par la présentation de l'unité centrale de notre système d'évaluation, le microprocesseur 6809, dont nous passerons en vue les diverses possibilités logicielles : des différentes catégories d'instructions aux modes d'adressage possibles.
Nous passerons ensuite à la présentation générale du système d'évaluation, du côté matériel, d'une façon succinte et précise puisque cela suppose que le lecteur ait déjà consulté le mémoire consacré à la conception et la réalisation matérielle du système.
Après cela, nous présenterons les différentes fonctions nécessaires à un système d'évaluation, et qui doivent être assumées par le programme moniteur. Toutes les routines seront présentées en détail, avec des organigrammes à l'appui, dans le but d'expliciter au maximum la méthode de mise en exécution de chaque tâche, ainsi que l'architecture globale du moniteur, c'est à dire la liaison entre les différents programmes utilitaires et le programme principal gérant l'ensemble et qui constitue le noyau du moniteur.

# II_ ORGANISATION du 6809

Le 6809 est un microprocesseur huit bits qui possède des caractéristiques introuvables sur ses prédécesseurs:
Le 6809 peut éxécuter des opérations sur des mots de seize bits; et bien qu'il ne dispose que d'un jeu d'instructions classique (59 types différents contre 72 pour le 6800), il comprend 1464 combinaisons différentes auxquelles sont associés ses divers modes d'adressages qui lui conférent une puissance et une facilité de programmation inouïes.

1- Architecture interne:

Le 6809 possède huit registres internes:

- Les accumulateurs A et B, de huit bits chacun, liés à l'UAL et qui peuvent être concaténés pour former un seul accumulateur de seize bits et réaliser des opérations sur des données de deux bytes.
- Les index X et Y, de seize bits, servant de registres d'adresse pour l'échange des données entre le microprocesseur et la mémoire.
- Les piles S et U, de seize bits, S est réservée pour la sauvegarde de l'état du processeur lors d'une interruption ou d'un passage vers un sous programme, tandis que U est laissée à la propre initiative du programmeur. Les registres S et U peuvent servir d'index au mime titre que X et Y.
- Le registre DP, de huit bits, qui contient le poids fort d'une

adresse lors d'un adressage direct.

- Le registre d'état _CCR_ qui contient des indicateurs définissant l'état du processeur après chaque opération.
- Le registre compteur _PC_ qui pointe toujours sur l'adresse mémoire devant être lue ou écrite.

## 2 - Instructions du 6809 :

Pour le 6809, une instruction contient deux champs :

- Le champ opération contenant le code qui renseigne sur le type de l'instruction et l'origine ou la destination des données à traiter.
- Le champ opérande contient la donnée même, ou bien l'endroit où elle doit être placée ou extraite.

Pour exécuter une instruction, le processeur procède en trois phases :

- 1ère phase : Recherche du code de l'instruction en mémoire.
- 2ème phase : Le code opération présent sur le bus de données est chargé dans le registre d'instruction du processeur puis décodé par son microprogramme interne.
- 3ème phase : Après recherche d'éventuelle opérande, le processeur exécute l'instruction.

Les instructions du 6809 se partagent en trois catégories principales :

a - Les instructions de transformation de données, qui regroupent :
- Les instructions arithmétiques.
- Les instructions logiques.
- Les instructions de comparaison et de test.

b- Les instructions de transfert de données qui incluent:

- Les instructions de chargement et de stockage.
- Les instructions de transfert interne au processeur.
- Les instructions d'empilement et de dépilement des piles.

c- Les instructions de saut et de branchement qui comprennent:

- Les instructions de branchement absolu.
- Les instructions de branchement relatif.
- Les instructions de saut et de retour de sous-programme.
- Les interruptions programmées.

3- Modes d'adressage:

La tâche principale d'un processeur est de traiter des données se trouvant en mémoire. Les modes d'adressage viennent définir l'origine de ces données pour compléter le rôle de chaque instruction.

La puissance d'un processeur est liée en grande partie à ses modes d'adressage. Le 6809 possède plus d'une dizaine de possibilitées pour spécifier l'opérande d'une instruction, ce qui lui confère une facilité accrue de mise au point d'un programme.

On distingue deux types de modes d'adressage:

- a- Modes d'adressage sans mémoire externe:

Dans ces modes, les instructions ne font intervenir que les registres internes du processeur ou des données implicites.

- Adressage Implicite:

Le champ opérande n'existe pas, toutes les informations nécessaires

pour l'exécution de l'instruction se retrouvent dans le code opération.

— Adressage par registres :

Le champ opérande contient le nom du ou des registres impliqués dans l'instruction.

exemple :

```
TFR   R1,R2
EXG   R1,R2
PSHS  A,B,X,CC
```

— Adressage immédiat :

Le champ opérande contient la donnée même à traiter par le processeur.

—b— Modes d'adressage avec mémoire externe :

— Adressage étendu :

Le champ opérande contient l'adresse effective de la donnée à traiter. Avec ce mode, on peut accéder à tout l'espace mémoire du processeur (les 64 K bytes).

— Adressage étendu indirect :

Cette fois, l'opérande contient non pas l'adresse d'une donnée, mais l'adresse de l'adresse de celle-ci.

exemple : LDA [$E000]

<$E000> = $10
<$E001> = $00

d'où : <A> = <$1000>

— Adressage direct :

Ce mode similaire au mode étendu, ne fournit que l'octet poids faible

de l'adresse mémoire à traiter. L'octet poids fort se trouve dans le registre DP. Ce mode, plus rapide que l'étendu et ne nécessitant qu'un octet pour une adresse, est conseillé lorsqu'on est en présence de tables de données à traiter se trouvant sur une seule page de 256 octets.

exemple: LDA $30 <DP> = $E0

<A> = <$E030>

— Adressages Indexés:

Dans ces modes, le processeur fait appel à une adresse de base située dans l'un de ses registres d'index X, Y, S, U. Et l'offset, s'il existe, indique le déplacement relatif à cette base en octets. L'addition de l'offset avec l'adresse de base fournit l'index qui sera l'adresse effective dans l'instruction. La nature et la longueur de l'offset va différencier les modes indexés entre eux. Nous trouverons:

— adressage indexé à offset constant nul ou sur 4 bits:

L'offset sera compris entre (-16 et +15), soit en hexadécimal (-$10 et $F).

Ce mode est le plus rapide des modes indexés et le moins encombrant en mémoire

exemple: LDA 0,X <X> = $1000 d'où <A> = <$1000>

LDA 3,X <X> = $1000 d'où <A> = <$1003>

— adressage indexé à offset sur 7 ou 15 bits:

Lorsque la valeur de l'offset dépasse l'intervalle [-16, +15], l'un de ces modes sera choisi.

Si l'offset est compris dans [-128, -17] ou [+16, +127], le déplacement sera sur 7 bits.

exemple: LDA 40,X

Si l'offset est compris dans [-32768, -129] ou [+128, +32767], le déplacement sur 15 bits sera choisi.

exemple : STA -400, X

— Adressage indexé à déplacement accumulateur A, B ou D :

Dans ce mode, l'offset est une variable signée contenue dans l'un des accumulateurs. Ce mode est surtout utilisée dans les conversions de codes.

exemples : LDA A, X <A> = $0F, <X> = $1000, d'où <A> = <$100F>

STA D, X <D> = $3000, <X> = $1000, d'où <$4000> = <A>

— Adressage indexé avec autoincrémentation ou autodécrémentation :

Ce mode est particulièrement utile lors du balayage d'une table de données, contenant des mots de même longueur. A chaque accès dans la table, on déplace l'index pour pointer sur le mot suivant.

L'index peut se déplacer dans les deux directions opposées :

- Des adresses basses vers les adresses supérieures, et l'index est incrémenté après chaque accès à l'adresse mémoire.
- Oubien des adresses hautes vers les adresses basses, mais cette fois-ci, l'index est décrémenté avant l'accès à la mémoire.

exemples : LDA ,X+ <X> = $E000 d'où <A> = <$E000> et <X> = $E001.

LDA ,-X d'où <A> = <$DFFF> et <X> = $DFFF

STA ,X++ <X> = $E000 d'où <$E000> = <A> et <X> = $E002

STA ,--X <X> = $DFFE et <$DFFE> = <A>

— Adressage indexé avec déplacement relatif au compteur programme :

Ce mode est particulièrement utile pour l'écriture de programmes

translatables, c'est à dire ne dépendant en aucune manière de l'espace mémoire où ils seront chargés.

Pour accéder à une adresse donnée, le processeur à la fin de lecture de l'instruction en cours, ajoute un offset au contenu du compteur programmes pour obtenir l'adresse effective à traiter. L'offset mesure la distance relative entre l'adresse de fin de l'instruction et l'adresse mémoire appelée par cette instruction. De cette façon, un programme ne comportera pas implicitement la valeur d'une adresse mémoire mais sa distance relative à l'instruction qui l'appelle.

-exemple:

Supposons que nous ayons à déplacer un programme stocké initialement à l'adresse $F800, vers l'adresse $2000. Et supposons que nous ayons à l'intérieur de ce programme des instructions en mode étendu de type: LDA $FA00. Pour déplacer ce programme, il faudrait rectifier cette instruction et mettre LDA $2200. Mais si plusieurs instructions de ce genre se trouvent mêlées dans le programme, sa correction devient onéreuse, et il s'avère interessant, dans cette situation, d'utiliser le mode indexé à déplacement relatif au PC.

Nous aurons: $F800 : LDA $FA00, PCR (A68D 01FC)

offset = adresse inscrite - adresse fin instruction.

= $FA00 - $F804 = $01FC.

En déplaçant cette instruction à l'adresse $2000:

$2000 : LDA $2200, PCR

A68D $01FC

l'adresse mémoire appelée sera = offset + adresse fin instruction.

= $01FC + $2004 = $2200.

— Modes indexés indirects:

Tous les modes indexés, à l'exception des modes à auto incrémentation et auto décrémentation d'une unité, acceptent le mode indirect.

-exemple: LDA [0,X]

<X> = $F000 , <$F000><$F001> = $3010 et <A> = <$3010>

LDA [,--U]

<U> = $3000 d'où <U> = $2FFE et <$2FFE><$2FFF> = $1000

d'où <A> = <$1000>

— Adressage relatif aux branchements:

Le 6809 possède trois types de branchements:

- Les branchements relatifs courts.
- Les branchements relatifs longs.
- Les branchements absolus.

-Les branchements Relatifs:

Ces branchements ont pour effet de changer la valeur courante du PC et d'interrompre ainsi l'exécution normale d'un programme pour la continuer à une adresse différente. Ils peuvent être conditionnels ou inconditionnels, et respectent la translatabilité d'un programme puisque l'opérande contient non pas l'adresse absolue du branchement mais sa distance relative à l'instruction en cours.

Pour les branchements courts, l'offset signé est compris entre 0 et 127 pour un déplacement positif, et entre -128 et -1 pour un déplacement négatif.

Pour les branchements longs l'offset signé est sur 15 bits, et permet donc d'accéder à toute adresse de l'espace mémoire du processeur.

_Les branchements absolus:

Ce sont les instructions JMP et JSR qui possèdent plusieurs modes d'adressage: l'étendu, le direct et l'indexé.

## 4- Système d'interruptions du 6809:

Le 6809 possède un système d'interruptions complet qui permet de résoudre bon nombre d'applications à temps réel.

Une interruption est une impulsion ou une instruction qui provoque l'arrêt d'un programme en cours d'exécution et le saut vers une séquence spéciale. Après exécution de celle-ci, le processeur reprend le programme interrompu en récupérant son contexte dans la pile du système.

Sur le 6809, on distingue trois types d'interruptions:

- Les interruptions matérielles.
- Les interruptions logicielles.
- Les interruptions matérielles programmées.

### _a- Interruptions matérielles:

Un niveau bas sur l'une de ces lignes d'interruption provoque l'arrêt du processeur et le saut vers le sous-programme d'interruption.

- Reset: Un niveau bas sur cette entrée provoque l'initialisation complète du processeur. Les registres sont mis à zéro, les interruptions sont interdites et le vecteur Reset est chargé dans le PC à partir des adresses $FFFE et $FFFF. Le programme Reset, se trouvant généralement en ROM, sert à démarrer un système lors d'une mise sous tension, et se termine par un branchement inconditionnel vers un moniteur résident.

- Interruptions NMI, IRQ, FIRQ:

Un niveau bas sur l'entrée NMI, arrête le programme en cours, et branche le processeur, après sauvegarde du contexte dans la pile, vers l'adresse contenue dans $FFFC et $FFFD. Cette interruption est non masquable et par conséquent on ne peut interdire son fonctionnement par programme. Par contre, les interruptions IRQ et FIRQ sont masquables. En activant l'une de ces lignes, le processeur teste d'abord le masque I ou F selon l'origine de la requête. Si ce masque est à un, le processeur ignore la demande d'interruption et continue normalement son programme. Sinon, il arrête son programme, sauvegarde l'ensemble de ses registres dans la pile pour une interruption IRQ, et met l'indicateur de sauvegarde totale E à 1. Puis saute vers le sous-programme d'interruption IRQ d'adresse contenue dans $FFF8 et $FFF9.

Pour la ligne FIRQ, qualifiée d'interruption rapide, le processeur, après test de masque, ne sauvegarde que le contenu du PC et du CCR dans la pile, et met l'indicateur E à 0 pour indiquer une sauvegarde partielle lors du dépilement. L'adresse de la routine d'interruption se trouve dans $FFF6 et $FFF7.

Chaque séquence d'exception doit se terminer par une instruction RTI, pour conduire le processeur à récupérer le contexte, du programme interrompu, stocké dans la pile.

- b. Interruptions logicielles: SWI, SWI 2, SWI 3.

Ce sont des instructions qui provoquent l'arrêt du programme les contenant; et de manière similaire aux interruptions matérielles, elles conduisent le processeur à se brancher vers une séquence spéciale.

Le processeur accorde une certaine priorité aux interruptions entre elles lorsque plus d'une requête est présente à la fois.
L'ordre de priorité est le suivant:

| | | |
|---|---|---|
| 1/ NMI | vecteur d'interruption: | FFFC et FFFD |
| 2/ SWI | | FFFA et FFFB |
| 3/ FIRQ | | FFF6 et FFF7 |
| 4/ IRQ | | FFF8 et FFF9 |
| 5/ SWI2 | | FFF4 et FFF5 |
| 6/ SWI3 | | FFF2 et FFF3 |

-c- Interruptions matérielles programmées:

CWAI : c'est une instruction d'attente d'interruption. Le processeur effectue un Et logique du mot d'état avec l'opérande immédiate pour valider l'une des interruptions masquables; puis sauvegarde l'ensemble de ses registres en attente d'interruption de la part de la ligne validée. CWAI permet de synchroniser le traitement d'une séquence spéciale avec l'arrivée du signal d'interruption, puisque la sauvegarde est déjà effectuée.

SYNC : A la rencontre de cette instruction, le processeur se met en attente d'interruption. Si l'interruption attendue est inhibée ou si le niveau bas dure moins de trois cycles horloges, le processeur continue l'exécution normale de l'instruction suivant SYNC. En réalité, SYNC sert à synchroniser le traitement d'un programme avec l'arrivée d'un évènement extérieur, sans orienter le processeur vers une interruption comme c'est le cas pour CWAI.

# III_ PRESENTATION du SYSTEME

Le système, construit autour du microprocesseur 6809, est destiné à recevoir et à traiter des programmes en langage hexadécimal, ainsi que toute application commandée par processeur. De ce fait, le système devra comprendre une forme d'entrées-sorties au minimum, de la mémoire vive et quelque interface pour d'éventuelle communication avec un autre système.

## 1- Synoptique du système:

Le système regroupe sur une seule carte:

- Le MPU 6809 avec son quartz.
- Deux K. octets de PROM contenant le programme moniteur.
- Un support pour une PROM supplémentaire.
- Deux K. octets de RAM.
- Un interface parallèle gérant le clavier et l'afficheur.
- Un interface parallèle supplémentaire.
- Un interface série à la norme RS-232C, pour d'éventuelles communications avec d'autres périphériques.

(voire Fig-1-)

## 2- Espace mémoire du système:

Après avoir défini les différentes parties que contiendra ce système, il est nécessaire de décider de certaines affectations de mémoire.

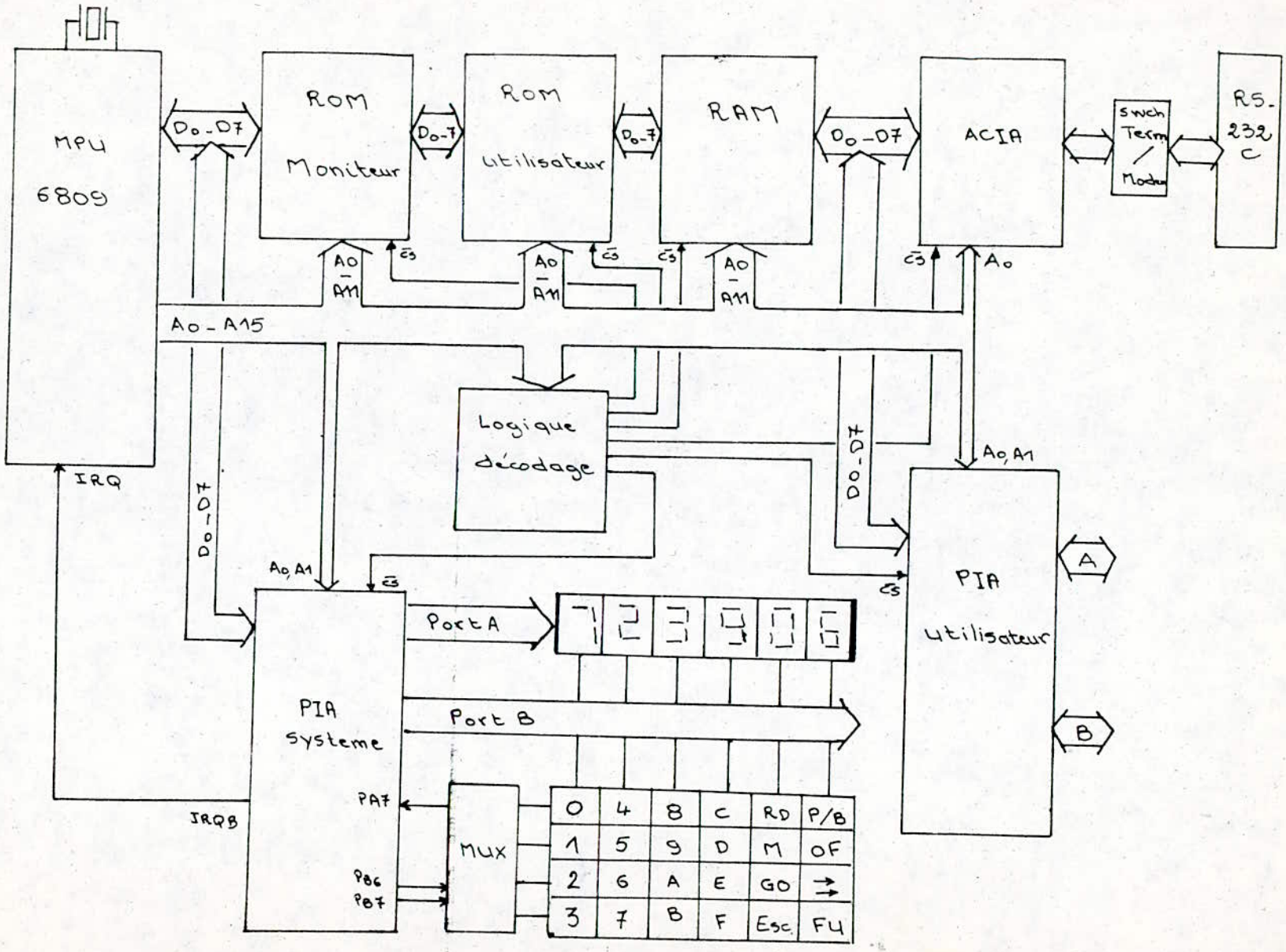
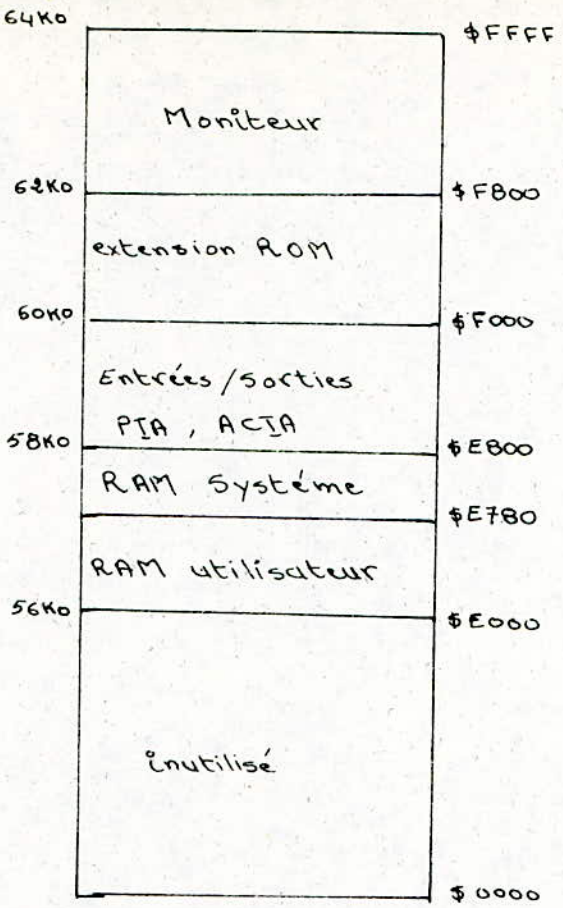

MPU
6809
IRQ
D0-D7
ROM
Moniteur
ROM
Utilisateur
RAM
ACIA
Switch
Term
/
Modem
RS-
232
C
A0
-
A11
CS
A0 - A15
Logique
décodage
A0, A1
PIA
Système
PIA
Utilisateur
PortA
Port B
IRQB
PA7
PB6
PB7
Mux
A
B
0 4 8 C RD P/B
1 5 9 D M OF
2 6 A E GO →
3 7 B F Esc FU

Fig-1-

La mémoire morte qui supporte le moniteur du systéme devra être située dans la partie la plus haute de l'espace d'adresses, soit de $F800 à $FFFF. Et ce, pour deux raisons :
Tout d'abord, le circuit de mise en route a pour effet de réveiller l'unité centrale avec son compteur programme initialisé à l'adresse $FFFE. Cette adresse doit contenir l'adresse de la partie du moniteur relative au démarrage du systéme, c'est à dire le programme RESET.
La seconde raison est que le systéme fonctionne et communique avec ses entrées-sorties par interruption. Les programmes relatifs au traitement des interruptions auront pour adresse, le contenu des adresses $FFF2 à $FFFD. De ce fait, le moniteur qui a pour fonction la réception et le traitement des requêtes de ses périphériques, devra contenir les vecteurs d'interruption pour aiguiller correctement le fonctionnement du systéme.

Pour le reste de l'espace mémoire du systéme, deux Kilo-octets seront réservés pour l'extension éventuelle du moniteur, ou l'implantation de programmes résidents par l'utilisateur. Suivi par une zone d'adresses des interfaces d'entrées-sorties. Ensuite, viendra l'espace de la mémoire vive. Nous avons jugé que, pour une programmation en langage machine, Deux Kilo-octets étaient largement suffisants. Les 128 octets supérieurs de la Ram seront réservés pour le systéme.
L'espace d'adressage restant est laissé à la propre initiative de l'utilisateur, qui peut étendre la capacité du systéme jusqu'à 58 K.octets, si toutefois son application particuliére l'exige.




Cartographie mémoire du système. Fig-2-

3- Gestion des entrées-sorties :

Pour notre système, le clavier en tant qu'entrée et l'afficheur en tant que sortie constituent les seules E/S élémentaires. Toutefois, la disponibilité d'un interface parallèle et d'un interface série permet de rajouter d'autres périphériques comme une imprimante, un terminal... Analysons, maintenant, comment nos périphériques élémentaires sont interfacés au système pour pouvoir établir les programmes qui les traitent.

-a- Le clavier :

La figure -3- représente l'interfaçage du clavier avec l'unité centrale.

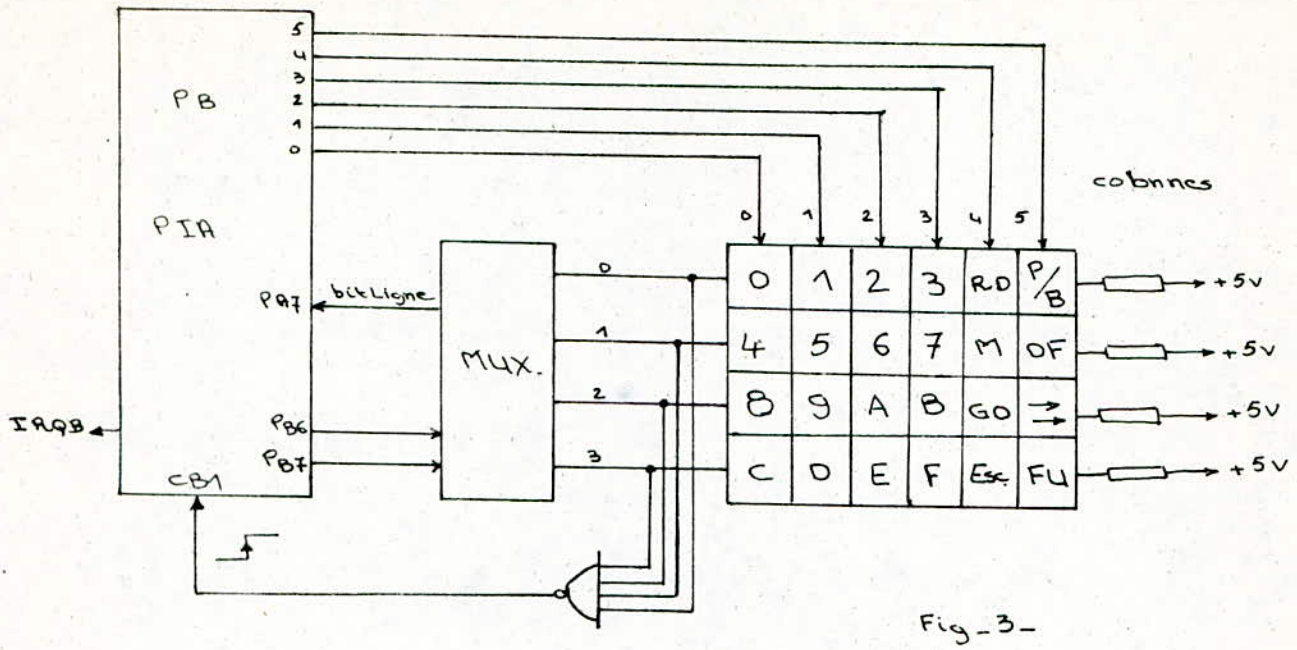
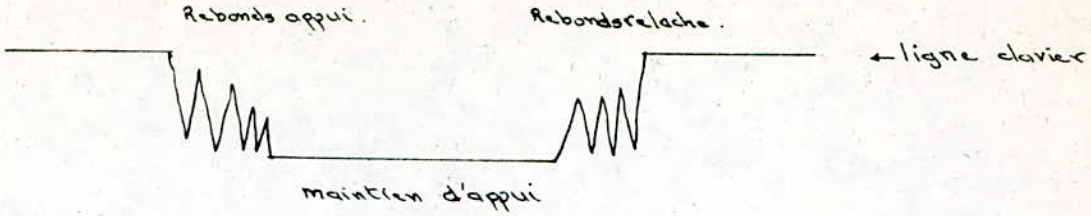



Fig_3_

Le clavier est organisé en matrice 4 x 6. Les colonnes sont reliées aux premières lignes du port B du PIA, les lignes quant à elles sont multiplexées et envoyées à travers la ligne PA7 au PIA, les lignes PB6 et PB7 sélectionnent par leur combinaison la ligne clavier sortante.

Le clavier fonctionne par interruption; les lignes initialement à l'état "1" forcent l'entrée CB1 du PIA à l'état bas. Celui-ci est programmé de façon à générer une interruption sur réception d'un front actif positif sur CB1. Les colonnes du clavier étant mises à l'état bas, l'appui sur une touche relie ligne à colonne correspondante et par conséquent un zéro fait basculer la porte Nand vers l'état haut qui envoie un front actif sur CB1.

Dans le traitement de l'interruption IRQ, le processeur viendra décoder la touche enfoncée par balayage des lignes puis balayage des colonnes. Mais avant cela, il doit éliminer les rebonds du clavier. Une touche appuyée ne se stabilise pas instantanément, de même qu'une touche relâchée; le contact fugitif rebondit un certain moment avant de se fixer.




Pour parer à cet inconvénient, il faut éviter de traiter la touche enfoncée durant ces rebonds, en temporisant le processeur pendant 25 ms, qui représentent le temps de stabilisation de la touche.

Les colonnes étant toujours à 0, le processeur balaye les lignes du clavier une à une par les lignes PB6 et PB7 du PIA. La ligne recherchée mettra le bit PA7 de ligne à l'état bas 0.

Le numéro de ligne trouvé, on maintient sur PA7 la ligne même, et on balaye les colonnes du clavier. La colonne recherchée est celle qui mettra la ligne trouvée à 0.

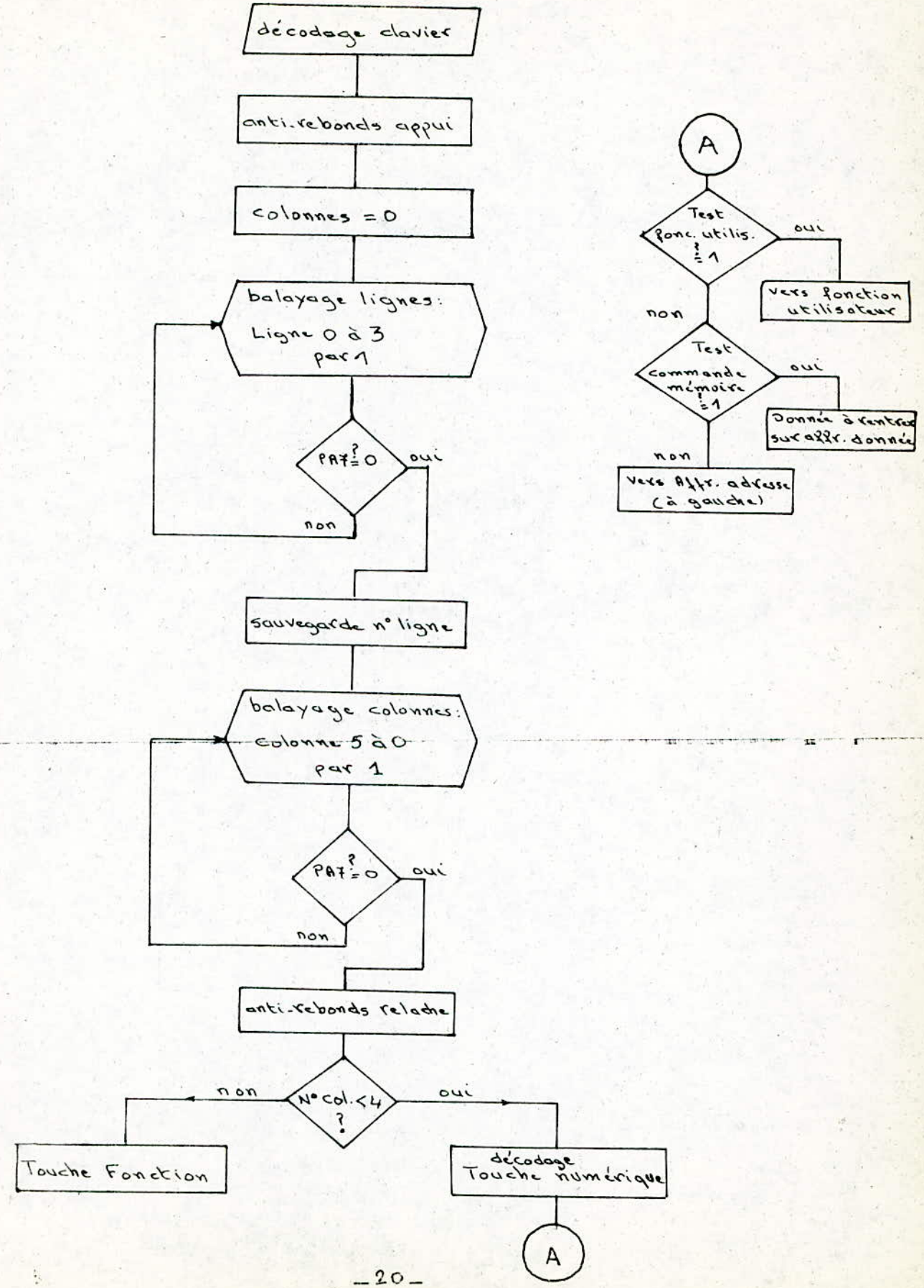
Pour une touche numérique, le numéro de colonne est inférieur à 4; nous pouvons donc, en combinant n° de ligne et n° de colonne, obtenir la valeur de la touche sans faire appel à une table de codage (voire organigramme).

Par contre une touche de commande sera codée entre $10 et $17, et son décodage fera appel à une table de branchements qui enverra le PC vers la routine correspondante à cette touche.

## -b- L'afficheur 6 chiffres.

La figure -4- représente l'organisation de l'afficheur de six chiffres.

Chaque groupe de deux chiffres est affecté à un octet de donnée. Pour visualiser simultanément ces 6 chiffres, il faudrait huit lignes pour chaque, soit 48 au total, ce qui est excessif. C'est pourquoi l'affichage multiplexé est utilisé, les afficheurs sont validés séquentiellement avec une fréquence de 1KHz pour tromper la persistence rétinienne.


décodage clavier
anti.rebonds appui
colonnes = 0
balayage lignes:
Ligne 0 à 3
par 1
PA7 ? = 0
oui
non
sauvegarde n° ligne
balayage colonnes:
Colonne 5 à 0
par 1
PA7 ? = 0
oui
non
anti-rebonds relache
N° col. < 4 ?
non
oui
Touche Fonction
décodage
Touche numérique
A
A
Test
Fonc. utilis.
? = 1
oui
vers fonction
utilisateur
non
Test
commande
mémoire
? = 1
oui
Donnée à rentrer
sur adr. donnée
non
vers Aff. adresse
(à gauche)

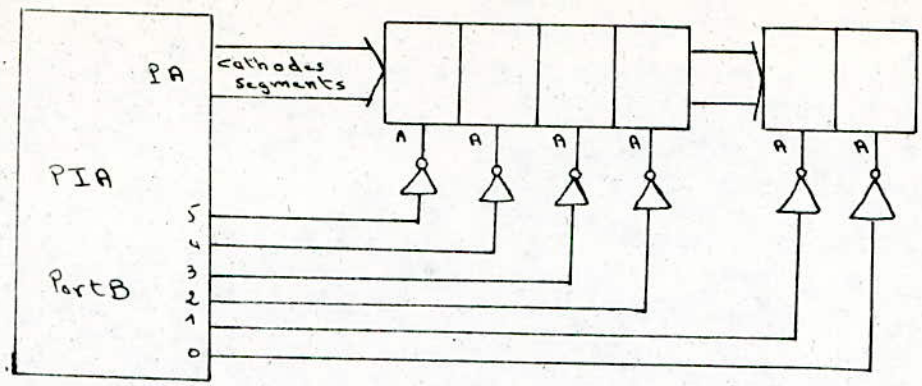
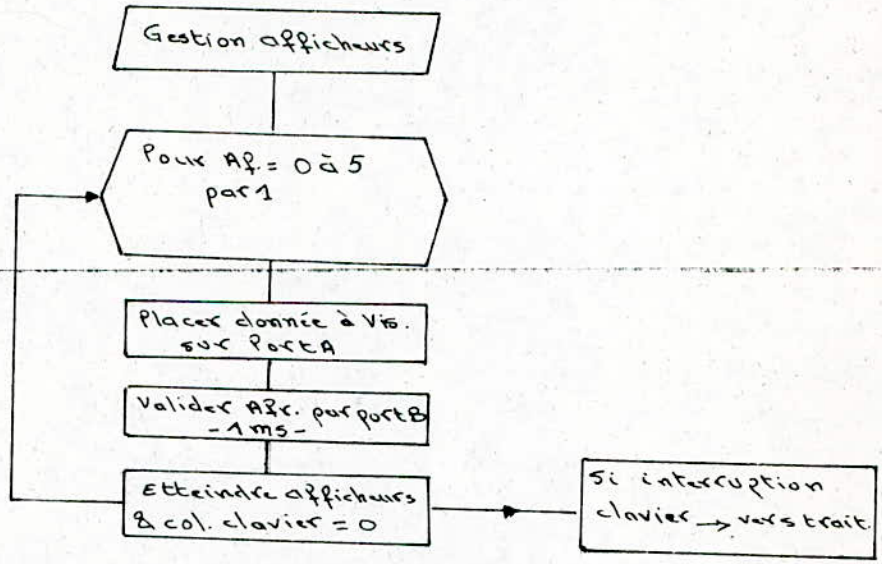



Fig-4-

Cet affichage multiplexé est géré par programme. Le port A du PIA véhicule l'information à visualiser pour chaque afficheur, le port B sélectionne un afficheur toutes les ms en polarisant correctement son anode commune.




Le programme de gestion des afficheurs est le programme principal du moniteur. Il reboucle à chaque fois qu'il n'y a pas d'interruption de la part du clavier. Et chaque routine spéciale se termine par un retour vers afficheurs pour visualiser les données traitées par le processeur.

Après avoir examiné les entrées. sorties du système, nous devons maintenant concevoir un langage de dialogue avec le programmeur. C'est l'objet du prochain chapitre.

# IV_ FONCTIONS du MONITEUR

Pour établir la liste des fonctions nécessaires au moniteur, il faut d'abord voire l'utilité du système réalisé et son domaine d'application.
Pour notre système, sa première application est destinée à l'enseignement de la programmation en assembleur du 6809. Il faudra donc, une possibilité d'entrée - sortie de données hexadécimales et une certaine facilité de mise au point et test de programmes écrits par l'utilisateur. Le moniteur comprendra donc toutes les fonctions d'un debugger dont voici la liste:

- Affichage du contenu d'une adresse mémoire.
- Possibilité de modification du contenu mémoire.
- Exécution d'un programme à partir d'une adresse spécifiée.
- Visualisation du contenu des registres du processeur après un programme.
- Possibilité de fixer le contenu de ces registres avant le lancement d'un programme.
- Balayage de l'espace mémoire par incrémentation ou décrémentation.
- Installation de points de coupure dans un programme.
- Exécution d'un programme instruction par instruction.
- Déplacement d'un bloc mémoire.
- Calcul d'offset pour les branchements relatifs.
- Possibilité de rajouter d'autres fonctions spéciales par l'utilisateur.

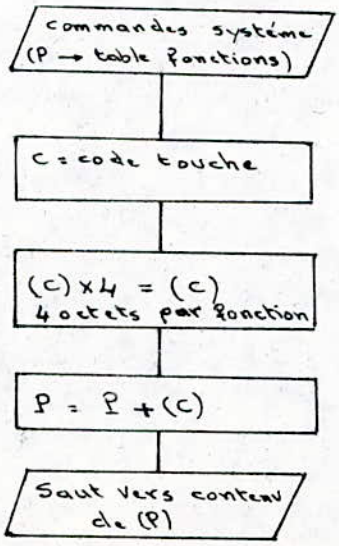
Toutes ces fonctions seront appelées ou lancées à partir du clavier. Pour cette raison, il est nécessaire de décider comment il doit être organisé.
Notre clavier possède 24 touches, 16 touches numériques et 8 pour les fonctions,

du moniteur. Or celles-ci excédent le nombre des touches disponibles. Il s'avère donc nécessaire de multiplexer l'emploi de certaines touches pour satisfaire toutes les tâches du moniteur.

-1- Organisation du clavier :

| | | | | | |
|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | RD $10 | P/B $14 |
| 4 | 5 | 6 | 7 | M $11 | oF $15 |
| 8 | 9 | A | B | GO $12 | →→ $16 |
| C | D | E | F | Esc. $13 | F4 $17 |

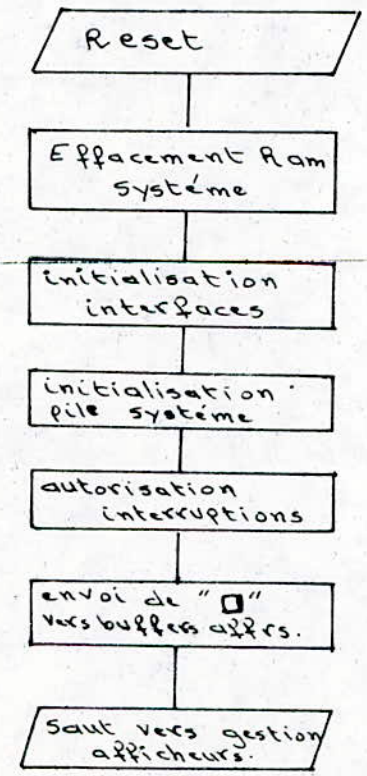
Le décodage des touches numériques est réalisé aisément; en combinant judicieusement n° de ligne et n° de colonne (Ligne = bits 2 et 3, colonne = bits 0 et 1) on obtient la valeur de la touche même. Pour les touches de commande, une conception fondée sur l'introduction d'une table, permet après décodage de sauter à la routine correspondant à la touche appuyée.




| codes | table fonctions |
|---|---|
| $10 | brancher vers Visu. Reg (4 octets) |
| $11 | brancher vers Visu. M. |
| $12 | Lancement programme |
| $13 | Retour moniteur |
| $14 | Breaks & Pas à pas |
| $15 | calcul offset. |
| $16 | déplacement programme |
| $17 | Fonctions utilisateur |

- 2 - Démarrage à froid du systéme (Reset) :

A la mise sous tension, la ligne Reset initialement à l'état bas, bascule vers l'état haut aprés quelques millisecondes, et entraine ainsi l'exécution de la routine de démarrage du systéme. Cette routine initialise correctement tous les interfaces servant l'échange entre l'utilisateur et le systéme, autorise les interruptions, puis passe la main au moniteur en affichant des " demi-zéros ". Le systéme est prêt maintenant à recevoir les requêtes de l'utilisateur. Cette séquence peut réintervenir à n'importe quel moment par simple appui sur le poussoir Reset, et cela pour recommencer de nouveau un travail mal exécuté ou pour retourner au moniteur en cas de perte de programme ou d'erreur de manipulation.



-3- Architecture du moniteur :

Le moniteur contient un programme principal qui s'exécute indéfiniment jusqu'à ce que une interruption l'arrête. c'est le programme de gestion des afficheurs ; et chaque séquence d'interruption une fois terminée renvoie le compteur programme vers la gestion des afficheurs. Il constitue donc le noyau du moniteur d'où et vers lequel transitent les routines spéciales (voire organigramme).

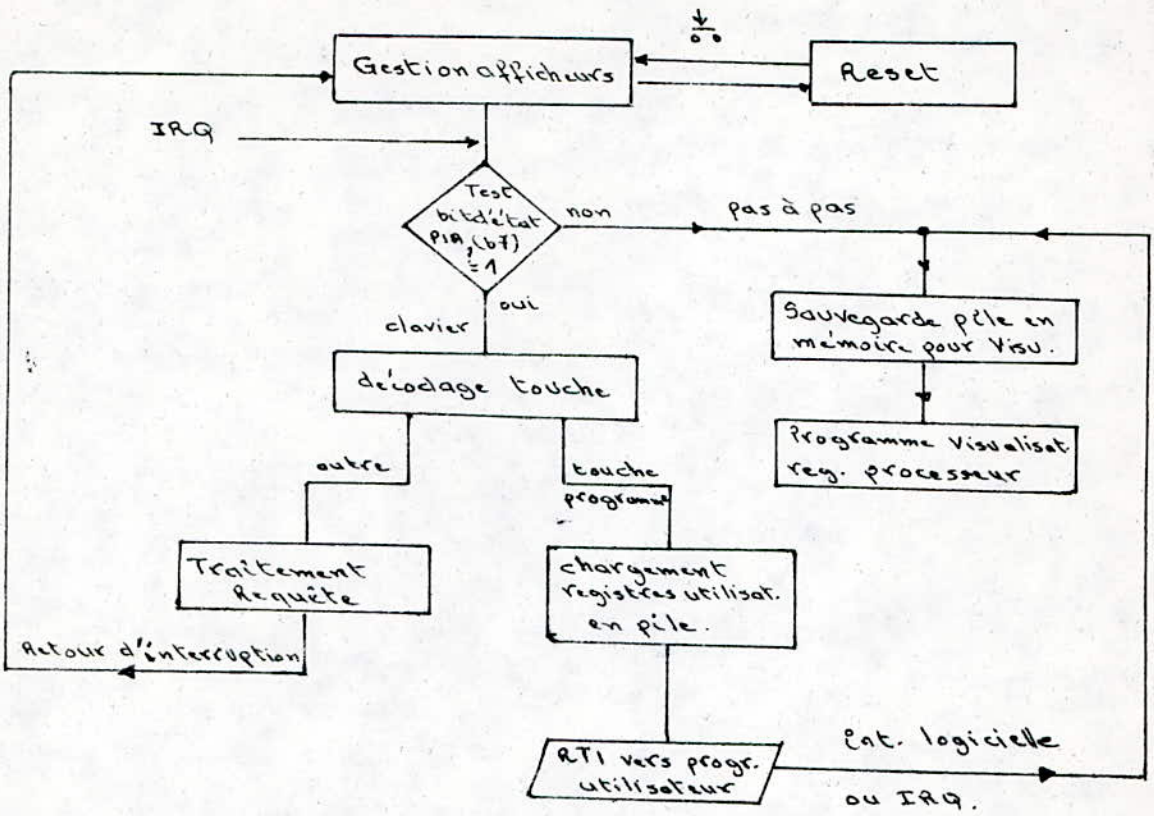
L'interruption matérielle utilisée dans le système est l'interruption masquable IRQ. Les deux autres NMI et FIRQ sont laissées au libre emploi de l'utilisateur. Un niveau bas sur la ligne IRQ provoque l'arrêt du moniteur et la sauvegarde de son contexte, puis le test de l'origine de la requête qui peut être soit le clavier ou bien le trace timer lors d'une exécution pas à pas.

- Pour le clavier, la touche enfoncée est décodée, traitée, puis repasse la main au moniteur pour visualiser un éventuel résultat. Pour une touche de lancement d'un programme, le retour d'interruption s'effectue avant l'exécution du programme en mettant dans la pile l'adresse du programme à exécuter comme contenu du PC.

- Si l'interruption provient du trace-timer, ou si nous sommes en présence d'une interruption logicielle, le contenu de la pile est copié dans zone mémoire en vue de le visualiser et de permettre ainsi au programmeur de suivre le déroulement de ses programmes, et de vérifier l'état du processeur après l'exécution.

-4- Fonctions "debugger" du moniteur :

Le moniteur comprend toutes les fonctions nécessaires à la mise au point et le test de programmes.



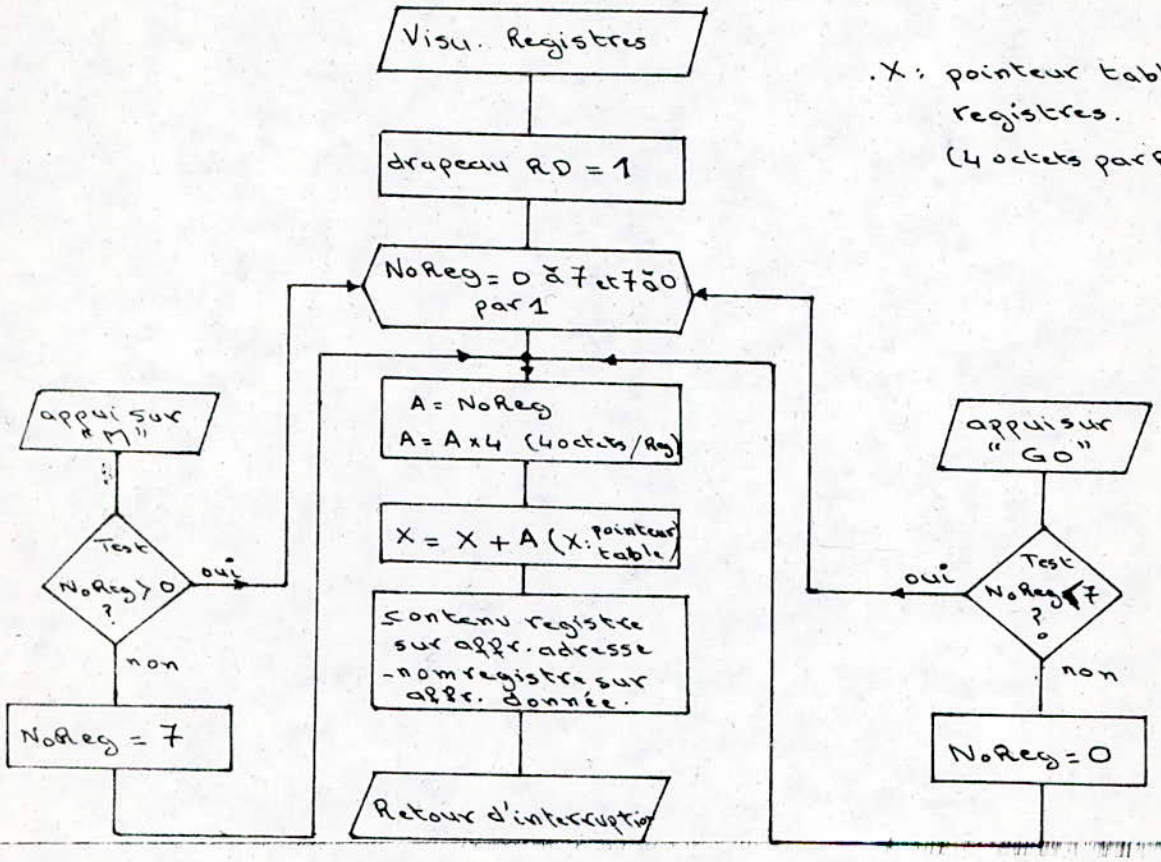

- Structure du moniteur -

A chaque fonction correspond une touche clavier, et comme chaque fonction exige l'emploi de plus d'une, beaucoup sont d'usage multiple. Pour déterminer à l'appui d'une touche sous quelle fonction elle devra opérer, des drapeaux seront testés. A chaque commande correspond un drapeau et celle qui est en cours d'exécution aura, au préalable, mis le sien à 1. Si la commande reboucle indéfiniment et ne passe pas la main au moniteur, pour en sortir l'utilisateur devra manœuvrer la touche "Escape" qui efface tous les drapeaux mis à un.

- a - Visualisation du contenu des registres du processeur:

Cette fonction, commandée par la touche "RD", permet d'examiner

le contenu des registres du processeur après l'exécution d'un programme spécial, et permet aussi d'enfixer le contenu avant le lancement d'un programme.

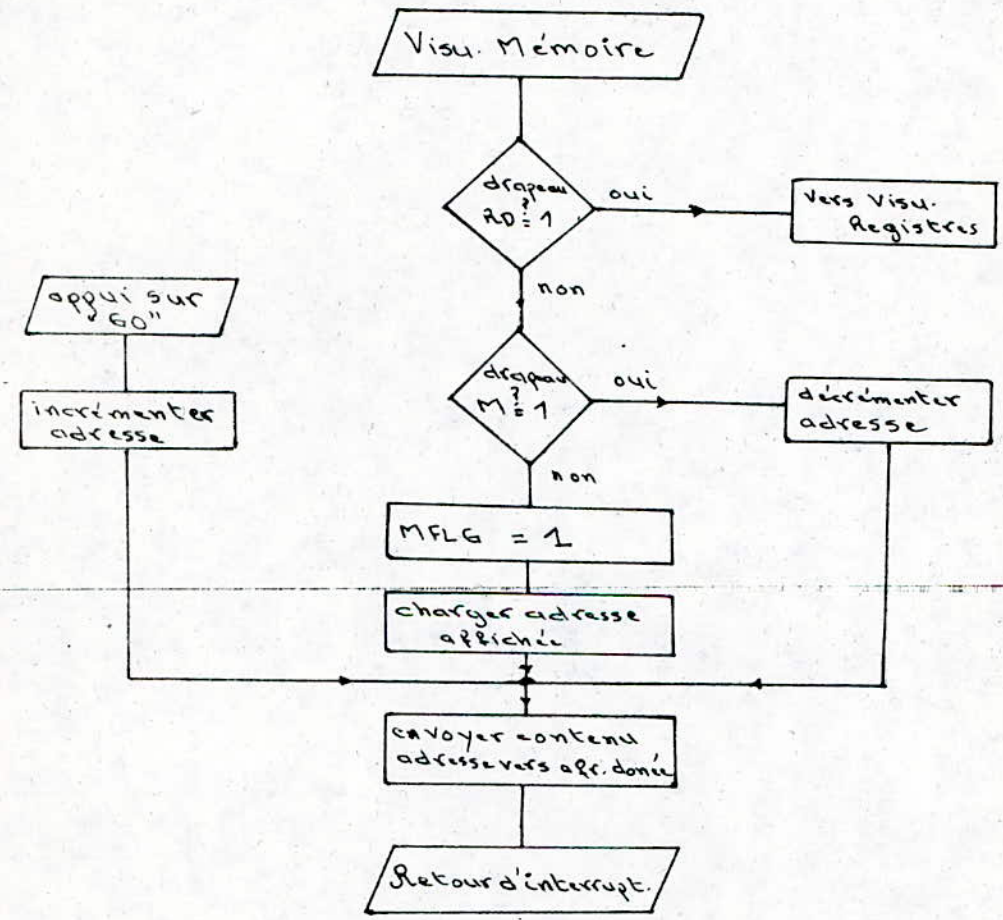



L'appui sur "RD" met le drapeau de la fonction à un, initialise un compteur de registres à 0, et positionne un pointeur sur la table des registres. Cette table contient 4 octets par registre, deux pour son contenu et deux autres pour son nom codé en code 7 segments.
Le pointeur de table incrémenté du compteur registres (multiplié par 4) pointe sur les données à afficher. Pour balayer tous les registres du processeur les touches "M" et "GO" permettent de passer respectivement, sous la fonction "RD", au prochain ou au précédent registre de la table.
Pour changer le contenu d'un registre, l'appui sur des touches hexadécimales

charge la nouvelle valeur affichée dans le registre même, en vue de récupération au moment du lancement d'un programme.

-b- Visualisation du contenu mémoire :

Cette fonction est la plus utilisée, elle permet au programmeur de vérifier le contenu d'une adresse mémoire et de rentrer ses propres programmes.




L'appui sur la touche "M" après un Escape ou un Reset, met le drapeau de la fonction à un, charge le contenu de l'adresse introduite précédemment sur le buffer de l'afficheur des données, et renvoie le programme vers la gestion des afficheurs après retour d'interruption.

Pour passer aux adresses supérieures ou inférieures, les touches "M" et "GO" servent, sous la commande 'M', à décrémenter ou incrémenter l'adresse de base introduite en premier lieu.
Pour changer le contenu d'une adresse mémoire (vive), un simple appui sur la valeur désirée change le contenu de l'adresse et l'affiche à droite.

-c- Calcul d'offset pour branchements relatifs:

En cas de branchement relatif, en introduisant l'adresse de l'offset et l'adresse de branchement, séparées par un appui sur la touche "OF", un simple appui sur la touche "GO" nous calcule cet offset et nous indique le type de branchement à choisir. Un second appui sur la même touche charge l'offset calculé dans l'adresse introduite au début du calcul (voire organigramme).

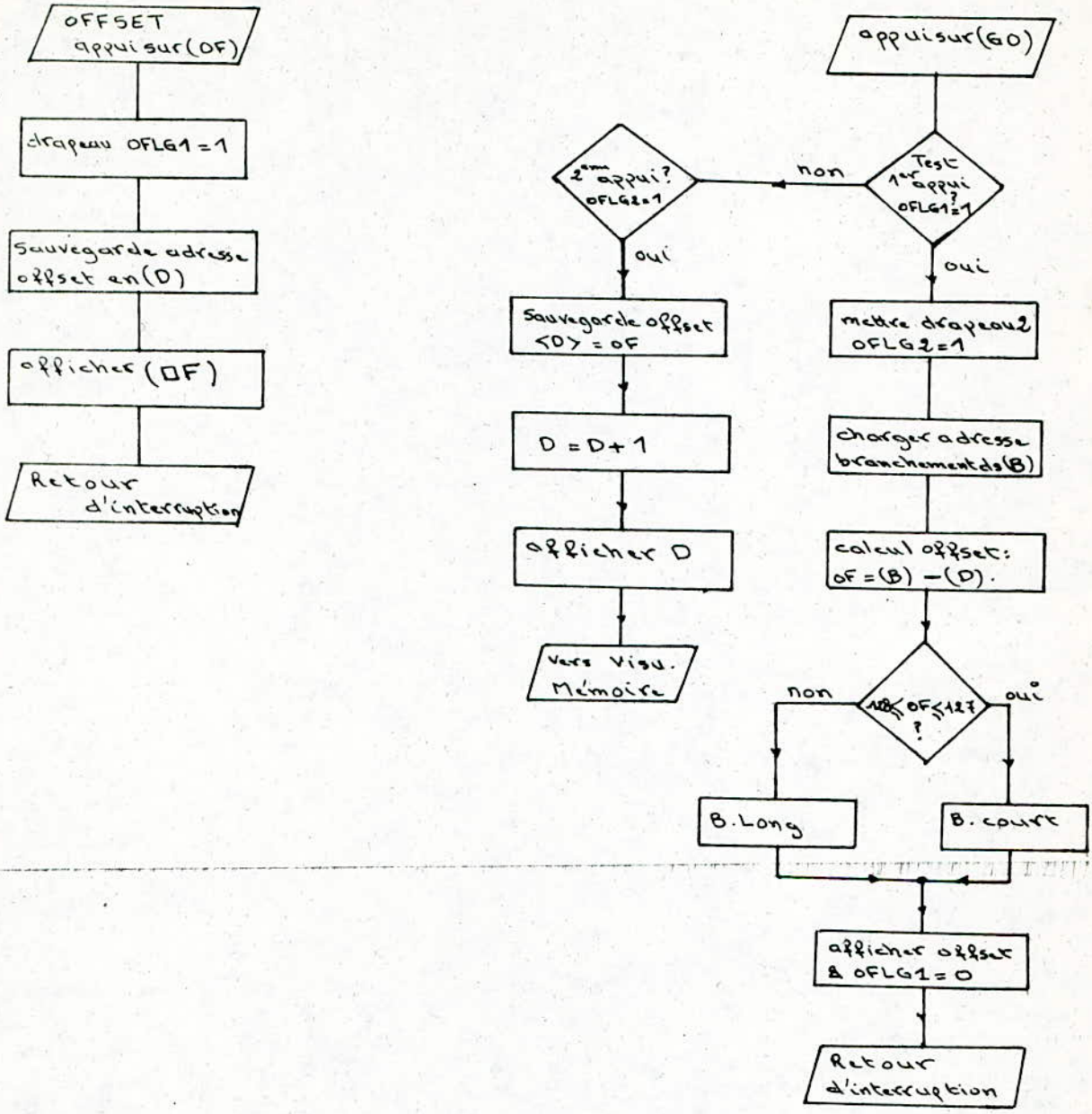
-d- Transfert d'une programme d'une zone vers une autre:

Dans le cas où l'on désire recopier un programme déjà chargé en mémoire, ou bien déplacer un programme gênant un autre, cette fonction s'avère très utile. En introduisant l'adresse de début et celle de fin de la séquence à transférer, séparées par l'appui sur la touche "MOVE", puis en introduisant l'adresse de transfert on lance le transfert en question par la touche "GO". Si l'adresse de transfert est à l'intérieur de la zone à déplacer, le transfert ne peut avoir lieu et le message d'erreur est affiché. (voire organigramme).

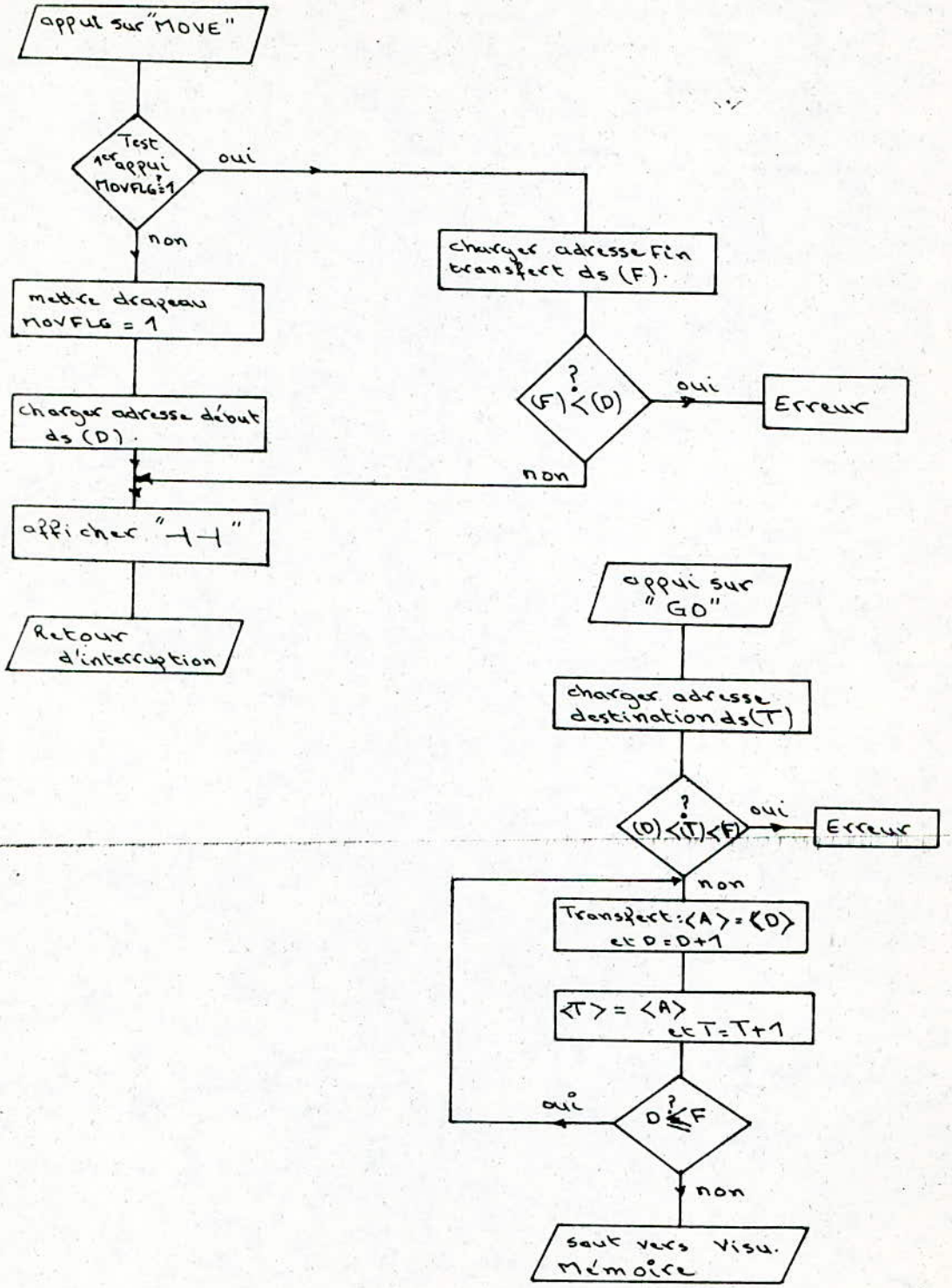
-e- Introduction de points d'arrêts et exécution pas à pas d'un programme:

Pour permettre au programmeur de suivre le déroulement de ses programmes, deux possibilités lui sont offertes:

- Soit l'exécution instruction par instruction avec retour sur




— Organigramme de la fonction "Calcul offset" —

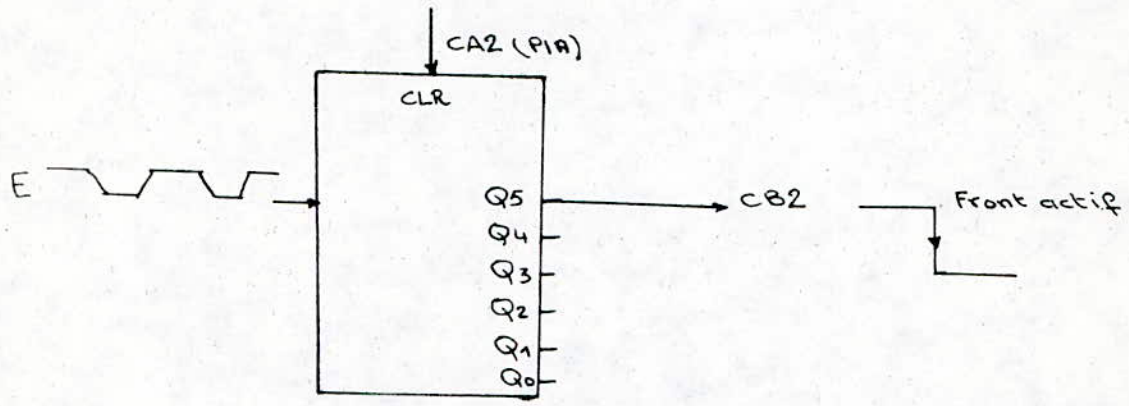



— Organigramme de la fonction "Transfert" —

visualisation des registres à chaque arrêt.

- Soit l'installation de points de coupure aux endroits jugés critiques par le programmeur.

La première approche consiste à initialiser un compteur binaire qui provoque l'interruption d'un programme juste au lancement de celui-ci.



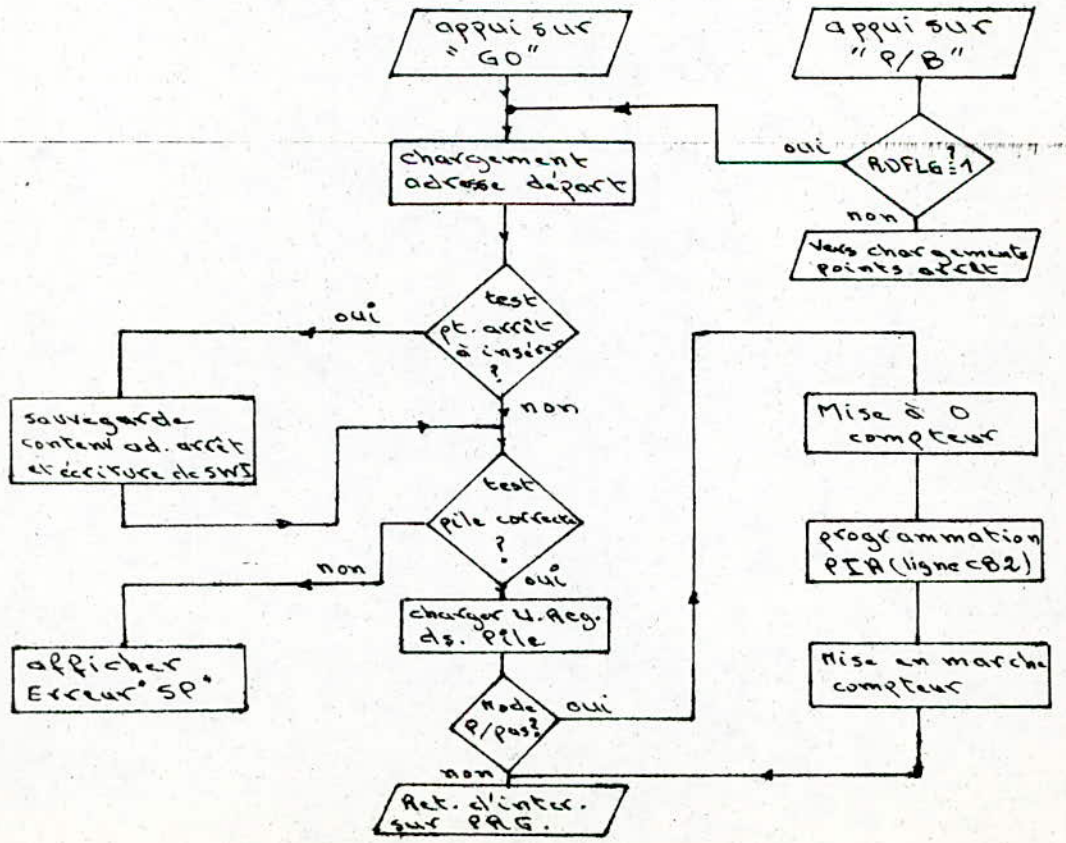

Le compteur opère de la façon suivante: juste avant le lancement du programme utilisateur, le compteur est mis à zéro, l'interface parallèle est programmé pour générer une interruption sur réception d'un front actif négatif sur la ligne CB2, puis le compteur est mis en marche à l'aide de la ligne CA2. Le front actif de CB2 n'interviendra que 32 cycles horloge plus tard, temps nécessaire au dépilement de la pile système et au lancement de la première instruction du programme.

Dans le cas où l'on ne désire vérifier le déroulement de son programme que sur certains points de test ou de branchement, la seconde approche s'avère utile. On place précédemment les adresses d'arrêts dans une table en mémoire à l'aide de la touche "P/B", et au moment du lancement d'un programme on insère le premier point d'arrêt en remplaçant le

contenu de l'adresse par le code de l'interruption logicielle SWI.
Le programme, à la rencontre de SWI, s'arrête, sauvegarde son contexte, récupère le contenu de l'adresse d'arrêt et saute vers la routine de visualisation des registres du processeur.

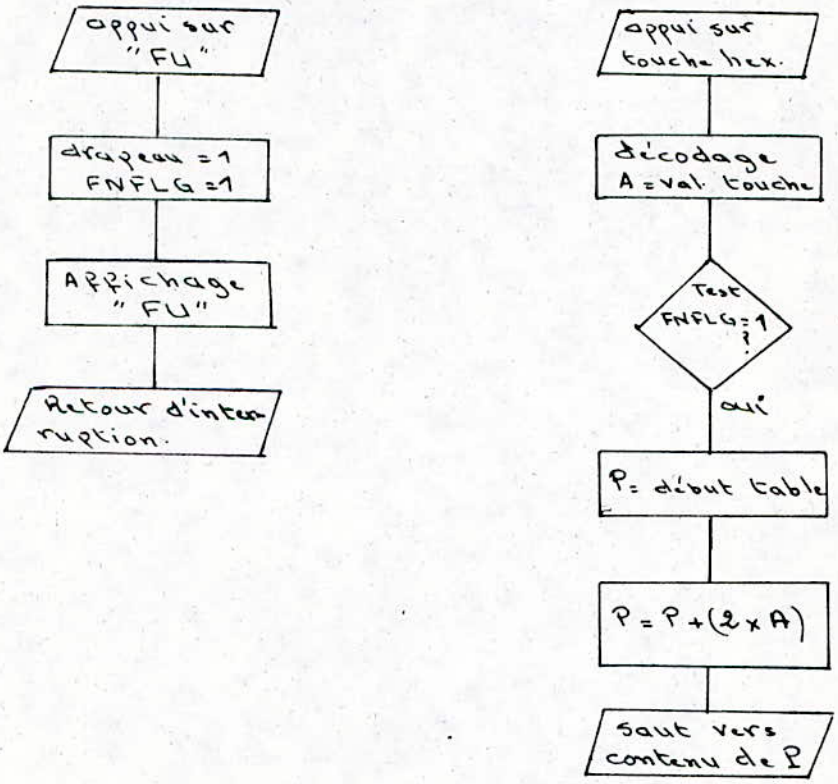
## -f- Lancement d'un programme pour exécution:

Pour exécuter un programme chargé en mémoire, on introduit l'adresse de départ et l'appui sur la touche "GO" charge cette adresse dans la pile du système avec le contenu des autres registres, et le lancement du programme s'effectue sur retour d'interruption.
Pour une exécution pas à pas, l'adresse de départ est introduite dans le PC utilisateur et l'appui sur "P/B" lance l'exécution avec arrêt après la première instruction.



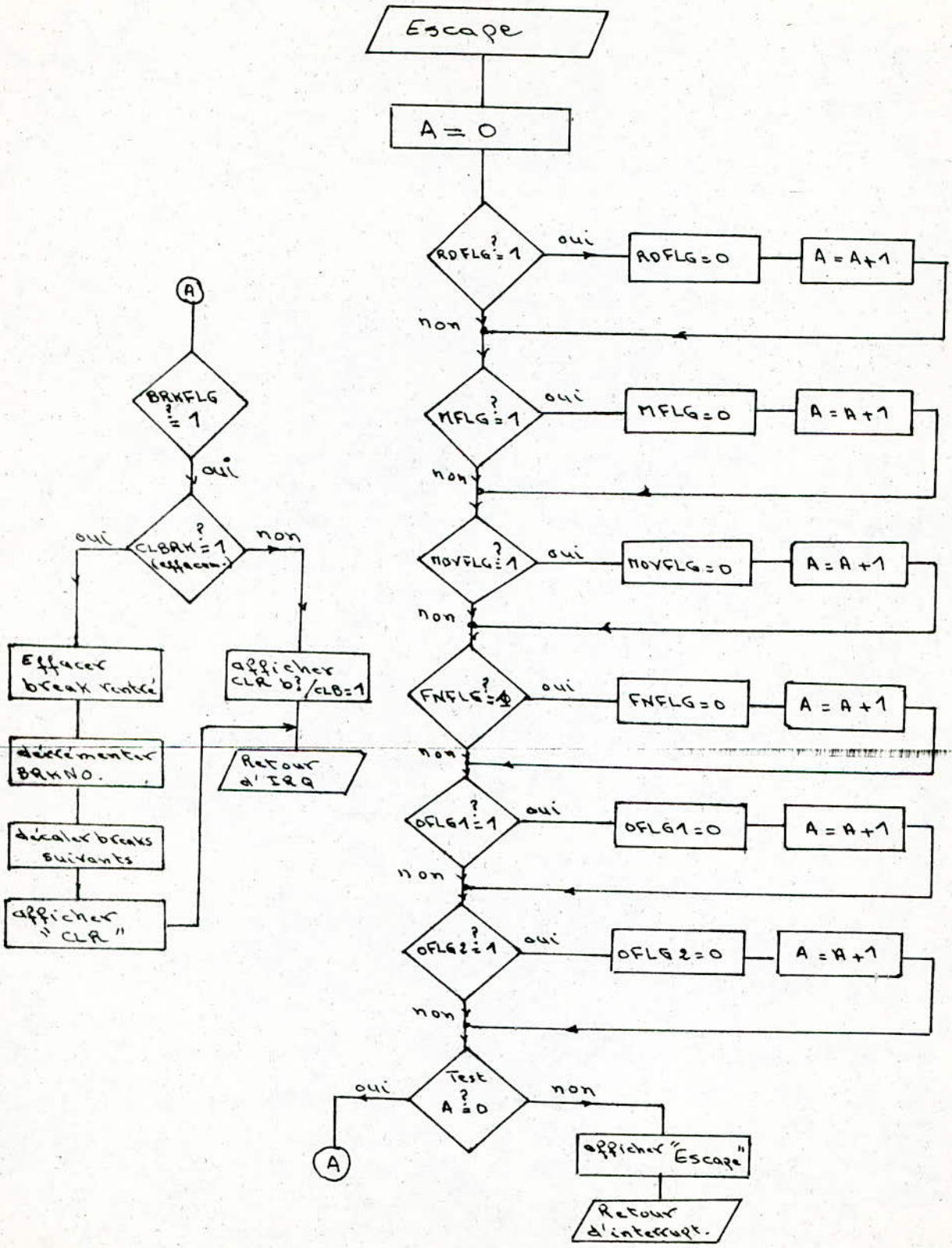
-g- Commandes utilisateur :

Si l'utilisateur désire rajouter d'autres fonctions, propres à son application particulière du système, une touche de déplacement permet d'appeler jusqu'à seize fonctions à l'aide du clavier.
La touche appel fonction utilisateur "FU" une fois activée met son drapeau à un. Et l'appui d'une touche numérique par la suite entraine le saut du moniteur vers une table contenant toutes les fonctions personnelles du programmeur et le branchement vers la fonction requise.




-h- Retour au moniteur :

Nous avons vu que chaque fonction, mettant son drapeau à un, monopolise d'autres touches fonctions pour son exécution.


Escape
A = 0
RDFLG ? = 1
oui
RDFLG = 0
A = A+1
non
MFLG ? = 1
oui
MFLG = 0
A = A+1
non
MOVFLG ? = 1
oui
MOVFLG = 0
A = A+1
non
FNFLG ? = 1
oui
FNFLG = 0
A = A+1
non
OFLG1 ? = 1
oui
OFLG1 = 0
A = A+1
non
OFLG2 ? = 1
oui
OFLG2 = 0
A = A+1
non
Test A ? = 0
oui
A
non
afficher "Escape"
Retour d'interrupt.
A
BRKFLG ? = 1
oui
CLBRK ? = 1 (effacem.)
oui
non
Effacer break rentré
décrémenter BRKNO.
décaler breaks suivants
afficher " CLR "
afficher CLR b? / CLB=1
Retour d'IRQ

Pour en sortir et pouvoir utiliser ces touches pour d'autres travaux, une touche "Escape" est prévue pour cet effet. L'appui sur cette touche efface tous les drapeaux mis préalablement à un. Un second appui sur cette touche l'initialise pour l'effacement d'éventuels points d'arrêts. Après l'affichage du message "CLR BREAKS ?", le programmeur introduit l'adresse d'arrêt qu'il ne désire plus utiliser, et appuie de nouveau sur la même touche. Le point d'arrêt est alors effacé de la table d'arrêts, le compteur d'arrêts décrémenté (5 arrêts maximum), et les points postérieurs sont décalés de deux cases mémoires pour occuper l'espace libéré par le point supprimé. L'opération se termine par l'envoi du message "CLR" qui rend la main au moniteur. (voire organigramme)

5 - Extension de l'emploi du système:

Ce système est conçu pour être fonctionnel sous sa forme minimale, avec son clavier comme entrée et son afficheur comme sortie. Toutefois, grâce aux interfaces d'entrées-sorties supplémentaires, le système peut être reconfiguré ou élargi en le reliant à d'autres périphériques.

Le 6809 possédant trois lignes d'interruptions différentes, chacune d'elles peut servir une tâche bien particulière. Le système peut, donc, communiquer avec son entourage par interruptions, sans qu'il y ait un conflit de priorité ou une quelconque ambiguité sur l'origine d'une interruption. Les vecteurs d'interruptions sont placés en mémoire vive grâce à un artifice de programmation. Et l'utilisateur n'aura plus qu'à charger l'adresse de sa routine d'interruption aux endroits indiqués.

# V - CONCLUSION

Ce système bati autour du microprocesseur 6809 est un système de faible cout, destiné à l'enseignement de la programmation en assembleur et à l'évaluation des capacités de toute la famille du microprocesseur.
Le 6809, contrairement à ses prédécesseurs, avec ses modes d'adressage variés, permet de concevoir et de développer des programmes bien structurés et peu avides d'espace mémoire.
Le moniteur, quant à lui, contient toutes les fonctions d'un debugger pour le test et la mise au point d'un programme.
Néanmoins, l'assemblage manuel du 6809 pose quelques difficultés, dans la mesure où les adressages indexés, qui constituent toute la puissance de programmation du processeur, exigent le calcul d'un post-octet. Le programmeur, pour éviter ce calcul, serait tenté de délaisser les modes indexés au profit des modes étendu et direct. Ce qui fait perdre au système tout son intérêt et toute sa particularité.
La conception et le romage d'un assembleur minimal, sur le système, permettront de parer à cet inconvénient et d'éviter tout manque d'exploitation des ressources du 6809.

# ANNEXES

CIT .AS: DEC 29,1986

```
00001                    ********************************************
00002                    *                                          *
00003                    *   MONITEUR du SYSTEME d'EVALUATION       *
00004                    *         a BASE du UP 6809                *
00005                    *                                          *
00006                    *   par: Mammar FODIL                      *
00007                    *      & Mahmoud LAMMALI                   *
00008                    *                                          *
00009                    *                -JANVIER 1987-            *
00010                    ********************************************

00012                    * DEFINITION des SYMBOLES *

00014          EFF0      PIAPA   EQU     $EFF0
00015          EFF0      PIADDA  EQU     $EFF0
00016          EFF1      PIACRA  EQU     $EFF1
00017          EFF2      PIAPB   EQU     $EFF2
00018          EFF2      PIADDB  EQU     $EFF2
00019          EFF3      PIACRB  EQU     $EFF3
00020          EFF8      ACIACR  EQU     $EFF8
00021          EFF8      ACIASR  EQU     $EFF8
00022          EFF9      ACIATD  EQU     $EFF9
00023          EFF9      ACIARD  EQU     $EFF9
00024          E7FE      SYSPIL  EQU     $E7FE
00025  E780                    ORG     $E780
00026  E780    0003      MHEX    RMB     $03
00027  E783    0006      MVISU   RMB     $06
00028  E789    0001      RDFLG   RMB     $01
00029  E78A    0001      MFLG    RMB     $01
00030  E78B    0001      MOVFLG  RMB     $01
00031  E78C    0001      FNFLG   RMB     $01
00032  E78D    0001      OFLG1   RMB     $01
00033  E78E    0001      OFLG2   RMB     $01
00034  E78F    0001      BRKFLG  RMB     $01
00035  E790    0001      BRKCPT  RMB     $01
00036  E791    0001      CLBRK   RMB     $01
00037  E792    0001      BCFLG   RMB     $01
00038  E793    0001      GOFLG   RMB     $01
00039  E794    0001      REGNO   RMB     $01
00040  E795    0001      REGNEW  RMB     $01
00041  E796    0001      PASS1   RMB     $01
00042  E797    0001      KEY     RMB     $01
00043  E798    0002      MEMDEB  RMB     $02
00044  E79A    0002      MEMFIN  RMB     $02
00045  E79C    0002      MEMSAV  RMB     $02
00046  E79E    0001      BRKNO   RMB     $01
00047  E79F    000A      BRKTBL  RMB     $0A
00048  E7A9    0005      BRKSAV  RMB     $05
```

:IT .AS: DEC 29,1986

```
)0050  E7AE  0001  UCC    RMB  $01
)0051  E7AF  0002  UD     RMB  $02
)0052  E7B1  0001  UDP    RMB  $01
)0053  E7B2  0002  UX     RMB  $02
)0054  E7B4  0002  UY     RMB  $02
)0055  E7B6  0002  UUS    RMB  $02
)0056  E7B8  0002  UPC    RMB  $02
)0057  E7BA  0002  USP    RMB  $02
)0058  E7BC  0002  FNTBL  RMB  $02
)0059  E7BE  0002  UNMI   RMB  $02
)0060  E7C0  0002  UFIRQ  RMB  $02


)0062                  * VECTEURS d'INTERRUPTIONS du 6809 *

)0064  FFF2                   ORG  $FFF2
)0065  FFF2  FAEC             FDB  SWINT3
)0066  FFF4  FAEA             FDB  SWINT2
)0067  FFF6  FAF2             FDB  FIRQ
)0068  FFF8  FA6E             FDB  IRQINT
)0069  FFFA  FABD             FDB  SWINT
)0070  FFFC  FAEE             FDB  NMI
)0071  FFFE  F800             FDB  RESET
```

KIT .AS: DEC 29,1986

```
00073  F800                        ORG     $F800
00074                        *********
00075                        * RESET *
00076                        *********
00077                        * Effacement Ram Systeme *
00078  F800 8E  E780        RESET  LDX     #$E780
00079  F803 6F  80          CLRST  CLR     ,X+
00080  F805 8C  E7FF               CMPX    #$E7FF
00081  F808 23  F9    F803         BLS     CLRST
00082                        *Initialisation interfaces *
00083  F80A 8E  EFF0               LDX     #$EFF0
00084  F80D 6F  80          CLREG  CLR     ,X+       Effacer Registres
00085  F80F 8C  EFF9               CMPX    #$EFF9    .
00086  F812 23  F9    F80D         BLS     CLREG     .
00087  F814 86  7F                 LDA     #$7F      PIA Systeme Direction:
00088  F816 B7  EFF0               STA     PIADDA    PortA en Sortie sauf B7
00089  F819 86  FF                 LDA     #$FF
00090  F81B B7  EFF2               STA     PIADDB    PortB en Sortie
00091  F81E 86  04                 LDA     #$04      PIA Registres Donnees
00092  F820 B7  EFF1               STA     PIACRA
00093  F823 B7  EFF3               STA     PIACRB
00094  F826 86  03                 LDA     #%00000011 Master Reset ACIA
00095  F828 B7  EFF8               STA     ACIACR
00096  F82B 10CEE7FE               LDS     #SYSPIL   Validation NMI
00097  F82F 10FFE7BA               STS     USP

00099                        *Visualisation "o" sur afficheurs*
00100  F833 8E  E783               LDX     #MVISU
00101  F836 86  5C                 LDA     #%01011100
00102  F838 A7  80                 STA     ,X+
00103  F83A 8C  E788               CMPX    #MVISU+5
00104  F83D 23  F9    F838         BLS     *-5
00105                        *Autorisation interruptions*
00106  F83F 1C  EF                 ANDCC   #%11101111 IRQ Valide
00107  F841 B6  EFF2               LDA     PIAPB     Effacer Bit7 du CRB
00108  F844 86  07                 LDA     #$07      Mode Interruption
00109  F846 B7  EFF3               STA     PIACRB    CB1 Actif Niveau Haut
00110  F849 7F  EFF2               CLR     PIAPB     Valider Lignes Clavier
00111  F84C 86  01                 LDA     #$01      1ere Passe Registres
00112  F84E B7  E795               STA     REGNEW    .
00113  F851 B7  E796               STA     PASS1
00114  F854 16  00B2  F909         LBRA    VISU      Vers Visualisation Affrs.
```

KIT .AS: DEC 29,1986

```
00117                          ******************************
00118                          * DECODAGE TOUCHES CLAVIER *
00119                          ******************************

00121  F857 7F  EFF2      KEYDEC CLR    PIAPB     Colonnes Clavier a 0
00122  F85A 17  0095 F8F2        LBSR   DLY25     Anti-Rebonds d'Appui
00123                     * Recherche Ligne  *
00124  F85D 8E  EFF0             LDX    #PIAPA
00125  F860 86  00               LDA    #%00000000 Ligne 0 ?
00126  F862 A7  02        LGNST  STA    2,X
00127  F864 6D  84               TST    ,X        PortA Bit7 Ligne MUX.
00128  F866 2A  06    F86E       BPL    LGNTRV    =0 :N. Ligne Trouve
00129  F868 8B  40               ADDA   #$40      sinon Ligne Suivante ?
00130  F86A 24  F6    F862       BCC    LGNST
00131  F86C 20  E9    F857       BRA    KEYDEC
00132  F86E 84  C0        LGNTRV ANDA   #%11000000 Sauvegarde N.Ligne
00133  F870 B7  E797             STA    KEY       en Memoire .
00134                     * Recherche Colonne *
00135  F873 86  1F.              LDA    #%00011111 Colonne 5 ?
00136  F875 C6  05               LDB    #$05      .
00137  F877 BA  E797      COLST  ORA    KEY       .
00138  F87A A7  02               STA    2,X       .
00139  F87C 6D  84               TST    ,X        BIT Ligne =0 ?
00140  F87E 2A  08    F888       BPL    COLTRV    OUI: N.Colonne trouve
00141  F880 84  3F               ANDA   #%00111111 NON: Voire Colonne suivante
00142  F882 44                   LSRA             .
00143  F883 5A                   DECB             .
00144  F884 2A  F1    F877       BPL    COLST     .
00145  F886 20  CF    F857       BRA    KEYDEC    Touche Desactivee !
00146  F888 6D  84        COLTRV TST    ,X
00147  F88A 2A  FC    F888       BPL    *-2
00148  F88C 8D  64    F8F2       BSR    DLY25     Anti-Rebonds Relache
00149  F88E 6D  84               TST    ,X        .
00150  F890 2A  C5    F857       BPL    KEYDEC    .
00151  F892 C1  04               CMPB   #$04      Touche Hex. ?
00152  F894 25  0D    F8A3       BLO    NUMKEY    oui
00153  F896 79  E797             ROL    KEY       sinon N.Ligne dans LSB
00154  F899 59                   ROLB             & N.Colonne dans Bit 2,3,4
00155  F89A 79  E797             ROL    KEY       .
00156  F89D 59                   ROLB             .
00157  F89E F7  E797             STB    KEY       .
00158  F8A1 20  21    F8C4       BRA    FONSEL    .
00159  F8A3 74  E797      NUMKEY LSR    KEY       Touche Hex. :
00160  F8A6 74  E797             LSR    KEY       N. Colonne dans bits 0,1
00161  F8A9 74  E797             LSR    KEY       N. Ligne dans bits 2,3
00162  F8AC 74  E797             LSR    KEY       .
00163  F8AF FA  E797             ORB    KEY       .
00164  F8B2 F7  E797             STB    KEY       .
```

```
0166  F8B5 7D  E78C           TST   FNFLG    Si Fonction Utilisateur ,
0167  F8B8 26  12   F8CC      BNE   USFNC    Brancher Table Utilisateur
0168  F8BA 7D  E78A           TST   MFLG     Si Routine Visu. Memoire ,
0169  F8BD 10260157 FA18      LBNE  ROLDTA   Vers Affrs. droite
0170  F8C1 16  0120 F9E4      LBRA  ROLADR   sinon vers Affrs. gauche


0172                        * SELECTION COMMANDES MONITEUR  *
0173                        * & COMMANDES UTILISATEUR       *

0175  F8C4 30  8DFFCA   FONSEL LEAX  SYSTBL-$40,PCR Pointer Table Commandes
0176  F8C8 58                 ASLB              4 Octets par Fonction
0177  F8C9 58                 ASLB               .
0178  F8CA 6E  85             JMP   B,X          .
0179  F8CC BE  E7BC     USFNC LDX   FNTBL    Table Utilisateur
0180  F8CF 58                 ASLB               .
0181  F8D0 6E  95             JMP   [B,X]
0182                        * Table Commandes Systeme *
0183  F8D2 6E  8D008E   SYSTBL JMP   REGPRG,PCR " RD "
0184  F8D6 6E  8D015B         JMP   MEMPRG,PCR " M  "
0185  F8DA 6E  8D0218         JMP   GOPRG,PCR " GO "
0186  F8DE 6E  8D041F         JMP   CLRPRG,PCR " Esc."
0187  F8E2 6E  8D02C6         JMP   BRKPRG,PCR " P/B "
0188  F8E6 6E  8D0325         JMP   OFPRG,PCR " OF "
0189  F8EA 6E  8D03B1         JMP   MOVPRG,PCR " ^ "
0190  F8EE 6E  8D0009         JMP   UFPRG,PCR "FU"

0192                        * Temporisation Rebonds Clavier *
0193  F8F2 108E186A     DLY25 LDY   #6250
0194  F8F6 31  3F            LEAY  -1,Y
0195  F8F8 26  FC   F8F6      BNE   *-2
0196  F8FA 39                 RTS


0198                        ***************************
0199                        * COMMANDES UTILISATEUR *
0200                        ***************************

0202  F8FB 86  01       UFPRG LDA   #$01      Mise a 1 Indicateur
0203  F8FD B7  E78C           STA   FNFLG      .
0204  F900 8E  713E         LDX   #$713E    Afficher "FU" a droite
0205  F903 BF  E787         STX   MVISU+4    .
0206  F906 16  0185 FA8E      LBRA  ENIRQ     & Retour d'Interruption
```

KIT .AS: DEC 29,1986

```
00208                          **********************************
00209                          * VISUALISATION SUR AFFICHEURS *
00210                          **********************************

00212  F909 8E  E783      VISU  LDX   #MVISU   Pointer Zone a Visualiser
00213  F90C C6  DF              LDB   #%11011111 1seul Affr. Allume,
00214  F90E 1A  01              ORCC  #%00000001 Carry a 1
00215  F910 A6  80        NEXAF LDA   ,X+
00216  F912 B7  EFF0            STA   PIAPA     vers Segments Affrs.
00217  F915 F7  EFF2            STB   PIAPB     Valider Afficheur
00218  F918 108E00F9            LDY   #249      Laisser 1ms
00219  F91C 31  3F              LEAY  -1,Y      .
00220  F91E 26  FC    F91C      BNE   *-2       .
00221  F920 86  00              LDA   #$00
00222  F922 B7  EFF0            STA   PIAPA     Etteindre Affrs.
00223  F925 B7  EFF2            STA   PIAPB     Colonnes Clavier a 0
00224  F928 56                  RORB             Afficheur Suivant
00225  F929 25  E5    F910      BCS   NEXAF
00226  F92B 20  DC    F909      BRA   VISU      Si 6eme Retourner au 1er

00228                          ************************
00229                          * DECODAGE 7 SEGMENTS *
00230                          ************************
00231  F92D 34  77        SEGCOD PSHS  U,Y,X,D,CC
00232  F92F 8E  E783            LDX   #MVISU
00233  F932 31  8D001E          LEAY  SEGTBL,PCR
00234  F936 CE  E780            LDU   #MHEX     Zone a Decoder
00235  F939 A6  C0        SEGST LDA   ,U+       Octet a Decoder
00236  F93B 1F  89              TFR   A,B
00237  F93D C4  0F              ANDB  #%00001111 Quartet bas dans B
00238  F93F 44                  LSRA
00239  F940 44                  LSRA
00240  F941 44                  LSRA
00241  F942 44                  LSRA             Quartet haut dans A
00242  F943 A6  A6              LDA   A,Y       Charger Code de A
00243  F945 E6  A5              LDB   B,Y       Charger Code de B
00244  F947 A7  80              STA   ,X+       Envoi vers Buffer Afficheur
00245  F949 E7  80              STB   ,X+
00246  F94B 1183E783            CMPU  #MHEX+3   Codage Termine ?
00247  F94F 25  E8    F939      BLO   SEGST     NON: Rprendre suite
00248  F951 35  77              PULS  U,Y,X,D,CC OUI: Retour Programme
00249  F953 39                  RTS

00251                          * TABLEAU CODE 7-SEGMENTS *

00253  F954      3F06      SEGTBL FDB   $3F06,$5B4F,$666D,$7D07
00254  F95C      7F6F             FDB   $7F6F,$777C,$395E,$7971
```

<IT .AS: DEC 29,1986

```
00256                          ****************************************
00257                          * VISUALISATION REGISTRES PROCESSEUR *
00258                          ****************************************
00259  F964 86  01        REGPRG LDA   #$01      Indiquer Visu. Registres,
00260  F966 B7  E789             STA   RDFLG     en Cours.
00261  F969 7F  E794             CLR   REGNO     1er Registre (PC)
00262  F96C B6  E794      REGDIS LDA   REGNO
00263  F96F CE  E783             LDU   #MVISU    Zone a Visualiser
00264  F972 30  8D004E           LEAX  REGTBL,PCR Table Registres
00265  F976 48                   ASLA
00266  F977 48                   ASLA             4 Octets par Registre
00267  F978 30  86               LEAX  A,X       Poiter sur Registre en cours
00268  F97A 7D  E795             TST   REGNEW    Est-ce 1ere Passe ?
00269  F97D 27  07    F986       BEQ   REGNAM    NON: Charger juste le Nom
00270  F97F EC  94               LDD   [,X]      OUI: Charger Contenu Registre
00271  F981 FD  E780             STD   MHEX
00272  F984 8D  A7    F92D       BSR   SEGCOD    Decodage 7-Segments

00274  F986 EC  02        REGNAM LDD   2,X       Charger nom du Registre
00275  F988 ED  44               STD   4,U       a Droite .
00276  F98A 30  94               LEAX  [,X]
00277  F98C C5  80               BITB  #$80      Registre 8 bits ?
00278  F98E 2A  06    F996       BPL   REGSTR
00279  F990 6F  C0               CLR   ,U+       si oui,Etteindre les 2
00280  F992 6F  C4               CLR   ,U        Affrs. de gauche .
00281  F994 20  05    F99B       BRA   UNOCT
00282  F996 B6  E780      REGSTR LDA   MHEX      Charger Contenu Registre
00283  F999 A7  84               STA   ,X        dans Registre meme.
00284  F99B 30  01        UNOCT  LEAX  1,X       Octet poids faible
00285  F99D B6  E781             LDA   MHEX+1
00286  F9A0 A7  84               STA   ,X
00287  F9A2 86  01               LDA   #$01
00288  F9A4 B7  E795             STA   REGNEW
00289  F9A7 16  00E4  FA8E       LBRA  ENIRQ     Retour d'Interruption

00291                          * Passage Registre Precedent *
00292  F9AA 7A  E794      REGDEC DEC   REGNO
00293  F9AD 2A  BD    F96C       BPL   REGDIS    si N. Registre >0,au Suivant
00294  F9AF 86  07               LDA   #$07      sinon Appeler CCR
00295  F9B1 B7  E794             STA   REGNO
00296  F9B4 20  B6    F96C       BRA   REGDIS
00297                          *
00298                          * Passage Registre Suivant *
00299  F9B6 B6  E794      REGINC LDA   REGNO
00300  F9B9 4C                   INCA
00301  F9BA 81  08               CMPA  #$08      si N.Registre <8,
00302  F9BC 25  01    F9BF       BLO   *+3       au Suivant.
00303  F9BE 4F                   CLRA             sinon Revenir a 0 (PC)
00304  F9BF B7  E794             STA   REGNO
00305  F9C2 20  A8    F96C       BRA   REGDIS
```

```
00307                          *
00308                          * Table Registres a Visualiser *
00309                          *
00310  F9C4  E7B8      REGTBL  FDB     UPC
00311  F9C6  73                FCB     %01110011,%00111001
00312  F9C8  E7AF              FDB     UD
00313  F9CA  00                FCB     %00000000,%01011110
00314  F9CC  E7B0              FDB     UDP-1
00315  F9CE  5E                FCB     %01011110,%11110011
00316  F9D0  E7B2              FDB     UX
00317  F9D2  06                FCB     %00000110,%01011110
00318  F9D4  E7B4              FDB     UY
00319  F9D6  06                FCB     %00000110,%01101110
00320  F9D8  E7B6              FDB     UUS
00321  F9DA  3E                FCB     %00111110,%01101101
00322  F9DC  E7BA              FDB     USP
00323  F9DE  6D                FCB     %01101101,%01110011
00324  F9E0  E7AD              FDB     UCC-1
00325  F9E2  39                FCB     %00111001,%10111001
```

```
<IT      .AS:    DEC 29,1986

)0328                          * RENTRER DONNEE HEX. dans  AFFICHEUR ADRESSE *

)0330  F9E4 B6  E797       ROLADR LDA     KEY       Valeur a Rentrer
)0331  F9E7 48                 ASLA
)0332  F9E8 48                 ASLA
)0333  F9E9 48                 ASLA
)0334  F9EA 48                 ASLA                  Sur Quartet Fort de A
)0335  F9EB C6  03             LDB     #03
)0336  F9ED 49                 ROLA                   par Permutation Circulaire
)0337  F9EE 79  E781           ROL     MHEX+1    Rentrer valeur de la Touche
)0338  F9F1 79  E780           ROL     MHEX      dans 4eme Affr.
)0339  F9F4 5A                 DECB
)0340  F9F5 2A  F6    F9ED     BPL     *-8
)0341  F9F7 17  FF33  F92D     LBSR    SEGCOD    Decodage 7-Segments
)0342  F9FA 7D  E789           TST     RDFLG     Mode Visu. Registres  ?
)0343  F9FD 26  08    FA07     BNE     *+10      oui : changement contenu Reg.
)0344  F9FF 7F  E787           CLR     MVISU+4
)0345  FA02 7F  E788           CLR     MVISU+5
)0346  FA05 16  0086  FA8E     LBRA    ENIRQ     sinon:  Retour d'Interruption
)0347  FA08 7F  E795           CLR     REGNEW    Indiquateur changement Reg.
)0348  FA0B B6  E794           LDA     REGNO
)0349  FA0E 81  06             CMPA    #$06
)0350  FA10 26  03    FA15     BNE     *+5
)0351  FA12 B7  E796           STA     PASS1
)0352  FA15 16  FF54  F96C     LBRA    REGDIS    Retourner vers Visu. Registres

)0354                          * RENTRER DONNEE HEX. dans AFFICHEURS DONNEES *

)0356  FA18 B6  E782       ROLDTA LDA     MHEX+2    Decalage a gauche
)0357  FA1B 48                 ASLA                  du contenu Affr.
)0358  FA1C 48                 ASLA
)0359  FA1D 48                 ASLA
)0360  FA1E 48                 ASLA
)0361  FA1F BA  E797           ORA     KEY       Valeur touche a rentrer
)0362  FA22 BE  E780           LDX     MHEX      Test Memoire a modifier:
)0363  FA25 8C  E780           CMPX    #$E780    Si dans Ram Systeme,
)0364  FA28 25  05    FA2F     BLO     *+7       Modification Interdite !
)0365  FA2A 8C  E7BB           CMPX    #$E7BB
)0366  FA2D 23  04    FA33     BLS     *+6
)0367  FA2F A7  9FE780         STA     [MHEX]    Sinon, changer contenu
)0368  FA33 20  15    FA4A     BRA     MEMCHG    & Retour vers Visu.Memoire
```

<IT .AS: DEC 29,1986

```
)0372                          ***********************************
)0373                          * VISUALISATION CONTENU MEMOIRE *
)0374                          ***********************************

)0376  FA35 7D  E789      MEMPRG TST    RDFLG    Mode Visu. Registres ?
)0377  FA38 1026FF6E F9AA        LBNE   REGDEC   oui:"M" decremente N.Reg.
)0378  FA3C 7D  E78A             TST    MFLG     Mode Visu. Memoire ?
)0379  FA3F 26  11    FA52       BNE    MEMDEC   oui:"M" decremente N.Adresse
)0380  FA41 86  01        MEMDIS LDA    #$01     sinon: mettre Indicateur de
)0381  FA43 B7  E78A             STA    MFLG     Visu. Memoire .
)0382  FA46 A6  9FE780    MEMDTA LDA    [MHEX]   Envoyer contenu memoire
)0383  FA4A B7  E782      MEMCHG STA    MHEX+2   vers Affrs. Donnees
)0384  FA4D 17  FEDD F92D        LBSR   SEGCOD   Decodage 7-Segments
)0385  FA50 20  3C    FA8E       BRA    ENIRQ    Retour d'Interruption

)0387                          * Passage Adresse Precedente *
)0388  FA52 7D  E781      MEMDEC TST    MHEX+1
)0389  FA55 26  03    FA5A       BNE    *+5
)0390  FA57 7A  E780             DEC    MHEX
)0391  FA5A 7A  E781             DEC    MHEX+1
)0392  FA5D 20  E7    FA46       BRA    MEMDTA

)0394                          * Passage Adresse Suivante *
)0395  FA5F B6  E781      MEMINC LDA    MHEX+1
)0396  FA62 81  FF               CMPA   #$FF
)0397  FA64 26  03    FA69       BNE    *+5
)0398  FA66 7C  E780             INC    MHEX
)0399  FA69 7C  E781             INC    MHEX+1
)0400  FA6C 20  D8    FA46       BRA    MEMDTA
```

IT .AS: DEC 29,1986

```
00403                          ******************************
00404                          * TRAITEMENT INTERRUPTIONS *
00405                          ******************************

00407  FA6E 7F  EFF0           IRQINT CLR    PIAPA     Etteindre Afficheurs
00408  FA71 8E  E783                  LDX    #MVISU    Effacer Buffer Affrs.
00409  FA74 6F  80                    CLR    ,X+       .
00410  FA76 8C  E788                  CMPX   #MVISU+5  .
00411  FA79 23  F9    FA74           BLS    *-5       .
00412  FA7B 86  04                    LDA    #$04      Interdire Interruptions
00413  FA7D B7  EFF3                  STA    PIACRB    durant Traitement de celle-ci.
00414  FA80 B6  EFF3                  LDA    PIACRB    Origine Requete ?
00415  FA83 2A  07    FA8C            BPL    *+9       .
00416  FA85 7D  E793                  TST    GOFLG     .
00417  FA88 1027FDCB  F857            LBEQ   KEYDEC    .
00418  FA8C 20  12    FAA0            BRA    TRACE     .

00420  FA8E B6  EFF2           ENIRQ  LDA    PIAPB     Effacement Bit7 du CRB
00421  FA91 86  07                    LDA    #$07      Autorisation Interruption
00422  FA93 B7  EFF3                  STA    PIACRB    Clavier .
00423  FA96 7F  EFF2                  CLR    PIAPB     Colonnes Clavier a 0
00424  FA99 A6  E4             RETINT LDA    0,S       Validation IRQ
00425  FA9B 84  EF                    ANDA   #%11101111
00426  FA9D A7  E4                    STA    0,S
00427  FA9F 3B                        RTI                Retour vers Gestion Affrs.

00429                          * Sauvegarde Registres pour Visualisation *
00430  FAA0 C6  0C             TRACE  LDB    #$0C
00431  FAA2 CE  E7BA                  LDU    #USP      Top Zone Sauvegarde
00432  FAA5 10EFC4                    STS    ,U        Sauvegarde contenu Pile
00433  FAA8 32  6C                    LEAS   12,S      Valeur Initiale Pile
00434  FAAA A6  E2                    LDA    ,-S       Copie Pile dans,
00435  FAAC A7  C2                    STA    ,-U       Zone Sauvegarde .
00436  FAAE 5A                        DECB
00437  FAAF 26  F9    FAAA            BNE    *-5
00438  FAB1 7F  E793                  CLR    GOFLG
00439  FAB4 30  8DFE51                LEAX   VISU,PCR  Retour d'Interruption
00440  FAB8 AF  6A                    STX    10,S      vers Gestion Affrs.
00441  FABA 16  FEA7  F964            LBRA   REGPRG    Aller vers Visu. Registres
```

```
00443                                  * Points d'Arret *
00444  FABD 7D  E78F        SWINT  TST     BRKFLG   PRG. avec Point Arret ?
00445  FAC0 27  DE    FAA0          BEQ     TRACE
00446                                  * Restauration du Contenu du Point d'Arret *
00447  FAC2 4F                       CLRA
00448  FAC3 8E  E7A9                 LDX     #BRKSAV  Table Sauvegarde Contexte
00449  FAC6 108EE79F                 LDY     #BRKTBL  Table Points d'Arret
00450  FACA 6D  6B                   TST     11,S     .
00451  FACC 26  02    FAD0           BNE     *+4      .
00452  FACE 6A  6A                   DEC     10,S     .
00453  FAD0 6A  6B                   DEC     11,S     Contenu PC au Point d'Arret
00454  FAD2 EE  A1                   LDU     ,Y++     .
00455  FAD4 11A36A                   CMPU    10,S     Recherche N.Point d'Arret
00456  FAD7 27  07    FAE0           BEQ     *+9      .
00457  FAD9 4C                       INCA             .
00458  FADA 81  05                   CMPA    #$05
00459  FADC 27  C2    FAA0           BEQ     TRACE    .
00460  FADE 20  F4    FAD4           BRA     *-10     .
00461  FAE0 E6  86                   LDB     A,X      Recuperation Contexte,
00462  FAE2 E7  F80A                 STB     [10,S]   d'Adresse Arret.
00463  FAE5 7C  E790                 INC     BRKCPT   Indicateur d'Arret Rencontre
00464  FAE8 20  B6    FAA0           BRA     TRACE    Vers Copie Pile dans USREG.

00466                                  * Interruptions Logicielles *
00467  FAEA 20  B4    FAA0 SWINT2    BRA     TRACE
00468  FAEC 20  B2    FAA0 SWINT3    BRA     TRACE

00470                                  * Interruptions Hardware Utilisateur *
00471  FAEE 6E  9FE7BE      NMI      JMP     [UNMI]   vers Sequence Utilisateur
00472  FAF2 6E  9FE7C0      FIRQ     JMP     [UFIRQ]  vers Traitement Utilisateur
```

KIT .AS: DEC 29,1986

```
00475                          ********************************
00476                          * LANCEMENT PROGRAMME : "GO"   *
00477                          ********************************

00479                          * Test Indicateurs Commandes *
00480  FAF6 7D  E789      GOPRG  TST    RDFLG     .
00481  FAF9 1026FEB9 F9B6        LBNE   REGINC    .
00482  FAFD 7D  E78A             TST    MFLG      .
00483  FB00 1026FF5B FA5F        LBNE   MEMINC    .
00484  FB04 7D  E78B             TST    MOVFLG    .
00485  FB07 102601C0 FCCB        LBNE   PRGTFR    .
00486  FB0B 7D  E78D             TST    OFLG1     .
00487  FB0E 1026011A FC2C        LBNE   OFCALC    .
00488  FB12 7D  E78E             TST    OFLG2     .
00489  FB15 10260169 FC82        LBNE   OFSTRE    .

00491  FB19 86  01               LDA    #$01
00492  FB1B B7  E793             STA    GOFLG
00493  FB1E 7D  E78F             TST    BRKFLG    Points d'Arret ?
00494  FB21 26  08    FB2B       BNE    INSBRK    oui: Inserer Premier
00495  FB23 BE  E780      GOIN   LDX    MHEX      Charger Adresse Depart,
00496  FB26 BF  E7B8             STX    UPC       dans PC .
00497  FB29 20  21    FB4C       BRA    LANCE     vers Lancement Programme
00498  FB2B 4F           INSBRK  CLRA              Initialiser Compteur
00499  FB2C 108EE79F             LDY    #BRKTBL   Table points d'Arret
00500  FB30 8E  E7A9             LDX    #BRKSAV   Table Sauvegarde Contenu
00501  FB33 FE  E780             LDU    MHEX      Point Depart User PRG.
00502  FB36 11A3A1       NEXINS  CMPU   ,Y++      Recherche 1er Arret,
00503  FB39 25  07    FB42       BLO    SAVE      Inclus dans PROG.
00504  FB3B 4C                   INCA              .
00505  FB3C 81  05               CMPA   #$05      .
00506  FB3E 27  E3    FB23       BEQ    GOIN      .
00507  FB40 20  F4    FB36       BRA    NEXINS    .
00508  FB42 E6  B3       SAVE    LDB    [,--Y]    Sauvegarde Contenu ADR. Arret
00509  FB44 E7  86               STB    A,X       .
00510  FB46 C6  3F               LDB    #$3F      Code de SWI charge dans ,
00511  FB48 E7  B4               STB    [,Y]      ADR. Arret .
00512  FB4A 20  D7    FB23       BRA    GOIN      vers Chargement PC
```

<IT .AS: DEC 29,1986

```
)0515  FB4C C6  0C        LANCE  LDB    #$0C     Nbr.octets Registres =12
)0516  FB4E CE  E7BA             LDU    #USP     Top Pile Utilisateur  ,
)0517  FB51 1F  41               TFR    S,X      Sauvegarde Pile Systeme
)0518  FB53 10EEC4               LDS    ,U       Pile Utilisateur
)0519                     * Verification Pile Systeme *
)0520  FB56 118CE780             CMPS   #$E780   Si Pile en ROM,
)0521  FB5A 25  08    FB64       BLO    *+10     ou sur bas RAM SYS.:Erreur
)0522  FB5C 118CE7D0             CMPS   #$E7D0   .
)0523  FB60 22  02    FB64       BHI    *+4      .
)0524  FB62 20  0A    FB6E       BRA    EROSP    .
)0525  FB64 86  E6               LDA    #$E6     .
)0526  FB66 34  02               PSHS   A        Pile Correcte ?
)0527  FB68 35  02               PULS   A        .
)0528  FB6A 81  E6               CMPA   #$E6     .
)0529  FB6C 27  0E    FB7C       BEQ    REGPUL   .

)0531  FB6E 7F  E793       EROSP  CLR    GOFLG    Erreur Pile :
)0532  FB71 1F  14               TFR    X,S      .
)0533  FB73 8E  6D73             LDX    #$6D73   Afficher "SP" a droite
)0534  FB76 BF  E787             STX    MVISU+4  .
)0535  FB79 16  0083  FBFF       LBRA   EROR     & Eror a gauche .
)0536  FB7C 7D  E796       REGPUL TST    PASS1
)0537  FB7F 26  02    FB83       BNE    *+4
)0538  FB81 32  6C               LEAS   12,S
)0539  FB83 7F  E796             CLR    PASS1
)0540  FB86 A6  C2               LDA    ,-U      Copie Registres User en Pile
)0541  FB88 A7  E2               STA    ,-S      .
)0542  FB8A 5A                   DECB            .
)0543  FB8B 26  F9    FB86       BNE    *-5      .
)0544  FB8D B6  E797             LDA    KEY      Test Mode P/pas
)0545  FB90 81  14               CMPA   #$14     .
)0546  FB92 1026FEF8  FA8E       LBNE   ENIRQ    .
)0547  FB96 86  3C               LDA    #$3C     Mise a 0 Compteur 5 Bits
)0548  FB98 B7  EFF1             STA    PIACRA   .
)0549  FB9B B6  EFF2             LDA    PIAPB    Effacer Bit7 d'interruption
)0550  FB9E 86  0E               LDA    #$0E     Interruption Valide sur
)0551  FBA0 B7  EFF3             STA    PIACRB   Front descendant de CB2
)0552  FBA3 86  34               LDA    #$34     Mise en marche Compteur
)0553  FBA5 B7  EFF1             STA    PIACRA   .
)0554  FBA8 12                   NOP             .
)0555  FBA9 16  FEED  FA99 PRGO  LBRA   RETINT   vers Retour d'Interruption
```

```
0558                          ****************************************
0559                          * POINTS d'ARRET :INSERTION & EXECUTION *
0560                          ****************************************

0562  FBAC 7D  E789      BRKPRG TST   RDFLG    Mode P/pas ?
0563  FBAF 27  09  FBBA         BEQ   *+11      .
0564  FBB1 7D  E790             TST   BRKCPT   Relancement d'1 PRG. avec
0565  FBB4 1026FF73 FB2B        LBNE  INSBRK   Points d'arret ?
0566  FBB8 20  92  FB4C         BRA   LANCE
0567  FBBA 7D  E78F             TST   BRKFLG   Suite Insertion Point Arret ?
0568  FBBD 26  0A  FBC9         BNE   NEXBRK   .
0569  FBBF 86  01               LDA   #$01     si 1ere fois , mettre
0570  FBC1 B7  E78F             STA   BRKFLG   Indicateur du Mode .
0571  FBC4 7F  E79E             CLR   BRKNO    Initialiser compteur
0572  FBC7 20  03  FBCC         BRA   *+5      .
0573  FBC9 7C  E79E      NEXBRK INC   BRKNO    .
0574  FBCC B6  E79E             LDA   BRKNO    .
0575  FBCF 81  05               CMPA  #$05     5 Points d'Arret Max.
0576  FBD1 27  2C  FBFF         BEQ   EROR     .
0577  FBD3 CE  E79F             LDU   #BRKTBL  Table Rangement Arrets
0578  FBD6 BE  E780             LDX   MHEX     Adresse Break
0579  FBD9 27  24  FBFF         BEQ   EROR     .
0580  FBDB 8C  E780             CMPX  #$E780   Autorise sur RAM User Seulement
0581  FBDE 24  1F  FBFF         BHS   EROR     .
0582  FBE0 48                   ASLA             .
0583  FBE1 4D                   TSTA              Breaks Rentres suivant,
0584  FBE2 26  11  FBF5         BNE   VERIFY   Ordre Croissant .
0585  FBE4 AF  C6        STBRK  STX   A,U      Memorisation Arret
0586  FBE6 44                   LSRA             .
0587  FBE7 B7  E782             STA   MHEX+2   & Affichage "b nh"
0588  FBEA 17  FD40 F92D        LBSR  SEGCOD   .
0589  FBED 86  7C               LDA   #$7C     .
0590  FBEF B7  E787             STA   MVISU+4  .
0591  FBF2 16  FE99 FA8E        LBRA  ENIRQ    Retour vers Afficheurs

0593                          * Verification de Succession Arrets dans l'ordre *
0594  FBF5 80  02        VERIFY SUBA  #$02
0595  FBF7 AC  C6               CMPX  A,U
0596  FBF9 23  04  FBFF         BLS   EROR
0597  FBFB 8B  02               ADDA  #$02
0598  FBFD 20  E5  FBE4         BRA   STBRK

0600                          * Envoi du Message d'Erreur *
0601  FBFF 8E  E783      EROR   LDX   #MVISU
0602  FC02 CC  7950             LDD   #$7950
0603  FC05 ED  81               STD   ,X++
0604  FC07 CC  5C50             LDD   #$5C50
0605  FC0A ED  81               STD   ,X++
0606  FC0C 16  FE7F FA8E        LBRA  ENIRQ
```

```
0608                        ***********************************
0609                        * CALCUL BRANCHEMENTS RELATIFS *
0610                        ***********************************

0612  FC0F 7F  E792         OFFPRG  CLR     BCFLG     Effacement Drapeaux
0613  FC12 7F  E78E                 CLR     OFLG2     .
0614  FC15 86  01                   LDA     #$01      Indiquer Calcul Offset en cours
0615  FC17 B7  E78D                 STA     OFLG1     .
0616  FC1A BE  E780                 LDX     MHEX      Adresse Offset
0617  FC1D BF  E79C                 STX     MEMSAV    .
0618  FC20 17  FD0A F92D            LBSR    SEGCOD    .
0619  FC23 8E  5C71                 LDX     #$5C71    Afficher "oF"
0620  FC26 BF  E787                 STX     MVISU+4   .
0621  FC29 16  FE62 FA8E            LBRA    ENIRQ     Retour d'Interruption

0623  FC2C 86  01           OFCALC  LDA     #01       Indiquer 1er appui sur "GO"
0624  FC2E B7  E78E                 STA     OFLG2     .
0625  FC31 7D  E781                 TST     MHEX+1    .
0626  FC34 26  03    FC39           BNE     *+5       .
0627  FC36 7A  E780                 DEC     MHEX      Calcul d'Offset
0628  FC39 7A  E781                 DEC     MHEX+1    .
0629  FC3C FC  E780                 LDD     MHEX      .
0630  FC3F B3  E79C                 SUBD    MEMSAV    .
0631  FC42 FD  E780                 STD     MHEX      Rangement en Buffer Affrs.
0632  FC45 24  04    FC4B           BCC     *+6       Type de Branchement ?
0633  FC47 81  FF                   CMPA    #$FF      .
0634  FC49 27  21    FC6C           BEQ     BCOURT    .
0635  FC4B 81  00                   CMPA    #00       .
0636  FC4D 26  03    FC52           BNE     *+5       .
0637  FC4F 5D                       TSTB              .
0638  FC50 2A  1A    FC6C           BPL     BCOURT    .
0639  FC52 7D  E781                 TST     MHEX+1    .
0640  FC55 26  03    FC5A           BNE     *+5       .
0641  FC57 7A  E780                 DEC     MHEX      .
0642  FC5A 7A  E781                 DEC     MHEX+1    .
0643  FC5D 17  FCCD F92D            LBSR    SEGCOD    .
0644  FC60 8E  7C38                 LDX     #$7C38    Long: Afficher "bL"
0645  FC63 BF  E787                 STX     MVISU+4   .
0646  FC66 7F  E78D         OFRET   CLR     OFLG1     .
0647  FC69 16  FE22 FA8E            LBRA    ENIRQ     Retour d'Interruption
0648  FC6C 17  FCBE F92D    BCOURT  LBSR    SEGCOD    Branchement Court:
0649  FC6F 86  01                   LDA     #$01      .
0650  FC71 B7  E792                 STA     BCFLG     .
0651  FC74 7F  E783                 CLR     MVISU     .
0652  FC77 7F  E784                 CLR     MVISU+1   Effacer Affrs. Inutilises
0653  FC7A 8E  7C39                 LDX     #$7C39    Afficher "bC"a Droite
0654  FC7D BF  E787                 STX     MVISU+4   .
0655  FC80 20  E4    FC66           BRA     OFRET     & retour d'interruption

0657                        * Stockage d'Offset en Memoire *
0658  FC82 BE  E79C         OFSTRE  LDX     MEMSAV    Adresse Offset
0659  FC85 FC  E780                 LDD     MHEX      Offset
```

```
:IT      .AS:    DEC 29,1986

)0661  FC8B 7D  E792          TST     BCFLG    BC ouBL ?
)0662  FC8B 26  04    FC91    BNE     ONEOCT   .
)0663  FC8D ED  81            STD     ,X++     BLONG :2 Bytes
)0664  FC8F 20  02    FC93    BRA     *+4      .
)0665  FC91 E7  80    ONEOCT  STB     ,X+      BCOURT :1 Byte
)0666  FC93 BF  E780          STX     MHEX     .
)0667  FC96 7F  E78E          CLR     OFLG2    .
)0668  FC99 7F  E792          CLR     BCFLG    .
)0669  FC9C 16  FDA2  FA41    LBRA    MEMDIS   Retour su  Visu. Memoire
)0670                         *------------------*

)0672                         ********************
)0673                         * COPIE PROGRAMME *
)0674                         ********************

)0676  FC9F 7D  E78B  MOVPRG  TST     MOVFLG   2eme Appui  r "MOVE" ?
)0677  FCA2 26  17    FCBB    BNE     OFENTR   .
)0678  FCA4 86  01            LDA     #$01     SINON: Indiq er Mouvement PRG.
)0679  FCA6 B7  E78B          STA     MOVFLG   .
)0680  FCA9 FC  E780          LDD     MHEX     Adresse Depar
)0681  FCAC FD  E798          STD     MEMDEB   .
)0682  FCAF 17  FC7B  F92D    MOVST   LBSR    SEGCOD   .
)0683  FCB2 CC  4646          LDD     #$4646   Afficher "__"
)0684  FCB5 FD  E787          STD     MVISU+4  .
)0685  FCB8 16  FDD3  FA8E    LBRA    ENIRQ    Retour d'Inter ption

)0687  FCBB FC  E780  OFENTR  LDD     MHEX     2eme Appui sur  ove"
)0688  FCBE 10B3E798          CMPD    MEMDEB   Adresse Fin < A  esse Début
)0689  FCC2 1025FF39  FBFF    LBLO    EROR     Sinon Erreur !
)0690  FCC6 FD  E79A          STD     MEMFIN   Sauvegarde Adre  e Fin PRG.
)0691  FCC9 20  E4    FCAF    BRA     MOVST    .

)0693  FCCB 10BEE780  PRGTFR  LDY     MHEX     Appui sur "GO" :
)0694  FCCF BE  E798          LDX     MEMDEB   Rentrer Adresse  ansfert
)0695  FCD2 10BCE798          CMPY    MEMDEB   Exterieure au PF  a Mouvoir ?
)0696  FCD6 23  06    FCDE    BLS     *+08     .
)0697  FCD8 10BCE79A          CMPY    MEMFIN   .
)0698  FCDC 23  0D    FCEB    BLS     *+15     Sinon Erreur !
)0699  FCDE FC  E79A          LDD     MEMFIN
)0700  FCE1 A3  84            SUBD    ,X       Longeur Programm  Copier
)0701  FCE3 E3  A4            ADDD    ,Y       .
)0702  FCE5 1083E780          CMPD    #$E780   Fin Copie dans R  Systeme ?
)0703  FCE9 25  03    FCEE    BLO     *+5
)0704  FCEB 16  FF11  FBFF    LBRA    EROR     Si oui Erreur
)0705  FCEE A6  80            LDA     ,X+      Transfert Byte pa  yte
)0706  FCF0 A7  A0            STA     ,Y+      .
)0707  FCF2 BC  E79A          CMPX    MEMFIN   .
)0708  FCF5 23  F7    FCEE    BLS     *-7      .
)0709  FCF7 7F  E78B          CLR     MOVFLG   Fin de Copie :
)0710  FCFA 10BFE780          STY     MHEX     Adresse Suivante S  Affrs.
)0711  FCFE 16  FD40  FA41    LBRA    MEMDIS   Vers Visu. Memoire
```

IT .AS: DEC 29,1986

```
0713                          ****************************************
0714                          * RETOUR MONITEUR & EFFACEMENT BREAKS *
0715                          ****************************************

0717                          * Sortie d'une Commande *
0718  FD01 8E  E789   CLRPRG  LDX   #RDFLG
0719  FD04 4F                 CLRA
0720  FD05 6D  84             TST   0,X       Commande "RD"en Cours ?
0721  FD07 27  05   FD0E      BEQ   *+7       .
0722  FD09 6F  84             CLR   0,X       Si oui,Effacer Drapeaux "RD"
0723  FD0B 6F  07             CLR   7,X       .
0724  FD0D 4C                 INCA             Indiquer Sortie de Commande
0725  FD0E 6D  01             TST   1,X       Commande "M" en Cours ?
0726  FD10 27  03   FD15      BEQ   *+5
0727  FD12 6F  01             CLR   1,X       Si oui Effacer Drapeau
0728  FD14 4C                 INCA             & Indiquer Sortie de Commande
0729  FD15 6D  02             TST   2,X       Commande "MOV" en Cours ?
0730  FD17 27  03   FD1C      BEQ   *+5
0731  FD19 6F  02             CLR   2,X       Si oui Effacer Drapeau
0732  FD1B 4C                 INCA
0733  FD1C 6D  03             TST   3,X       Commande "FU" en Cours ?
0734  FD1E 27  03   FD23      BEQ   *+5
0735  FD20 6F  03             CLR   3,X       Si oui Effacer Drapeau
0736  FD22 4C                 INCA             .
0737  FD23 6D  04             TST   4,X       Commande "OF" en Cours ?
0738  FD25 27  03   FD2A      BEQ   *+5       .
0739  FD27 6F  04             CLR   4,X       Si oui Effacer Drapeau
0740  FD29 4C                 INCA             .
0741  FD2A 6D  05             TST   5,X       Commande "GO" en Calcul Offset
0742  FD2C 27  05   FD33      BEQ   *+7       .
0743  FD2E 6F  05             CLR   5,X       Si oui Effacer Drapeau
0744  FD30 6F  0A             CLR   10,X      .
0745  FD32 4C                 INCA             .
0746  FD33 6F  08             CLR   8,X
0747  FD35 4D                 TSTA             Test Mode de Touche "C/FU"
0748  FD36 27  18   FD50      BEQ   BRKTST    Si Indicateur Nul,Voire Breaks

0750  FD38 7F  E780   ESCP    CLR   MHEX       Sinon Mettre Buffer Affrs. a C
0751  FD3B 7F  E781           CLR   MHEX+1     .
0752  FD3E 7F  E782           CLR   MHEX+2     .
0753  FD41 CC  796D           LDD   #$796D     Afficher "ESCP"
0754  FD44 FD  E783           STD   MVISU      .
0755  FD47 CC  3973           LDD   #$3973     .
0756  FD4A FD  E785           STD   MVISU+2    .
0757  FD4D 16  FD3E FA8E      LBRA  ENIRQ      .
```

IT .AS: DEC 29,1986

```
0760                            * Traitement Points d'Arret *
0761  FD50 6D  06        BRKTST  TST     6,X       Points d'Arret ?
0762  FD52 27  E4  FD38           BEQ     ESCP      .
0763  FD54 6D  08                 TST     8,X       Si oui,1er Appui sur "Esc." ?
0764  FD56 26  16  FD6E           BNE     EFFAC     Non,Aller Effacer Break
0765  FD58 CE  E783               LDU     #MVISU    Oui: Afficher "CLR B?"
0766  FD5B CC  7C53               LDD     #$7C53    .
0767  FD5E ED  44                 STD     4,U       .
0768  FD60 CC  3938       RET     LDD     #$3938    .
0769  FD63 ED  C4                 STD     ,U        .
0770  FD65 C6  50                 LDB     #$50      .
0771  FD67 E7  42                 STB     2,U       .
0772  FD69 6C  08                 INC     8,X       & Indiquer 2eme Appui
0773  FD6B 16  FD20 FA8E          LBRA    ENIRQ     .
0774  FD6E BE  E780       EFFAC   LDX     MHEX      Si Buffer Affrs.=0,
0775  FD71 27  C5  FD38           BEQ     ESCP      Pas de Points d'Arret a Efface
0776  FD73 5F                     CLRB               Sinon:
0777  FD74 CE  E79F               LDU     #BRKTBL   Recherche Break dans Table
0778  FD77 AC  C5                 CMPX    B,U       .
0779  FD79 27  09  FD84           BEQ     *+11      .
0780  FD7B 5C                     INCB               .
0781  FD7C 5C                     INCB               .
0782  FD7D C1  0A                 CMPB    #$0A      .
0783  FD7F 25  F6  FD77           BLO     *-8       .
0784  FD81 16  FE7B FBFF          LBRA    ERQR      Si Introuvable :Erreur!
0785  FD84 7A  E79E               DEC     BRKNO     .
0786  FD87 2A  0B  FD94           BPL     NEXDEC    .
0787  FD89 7F  E78F               CLR     BRKFLG    Si Break Final Effacer
0788  FD8C 7F  E791               CLR     CLBRK     Drapeaux.
0789  FD8F CE  E783               LDU     #MVISU
0790  FD92 20  CC  FD60           BRA     RET       .
0791  FD94 1F  98         NEXDEC  TFR     B,A       Decalage Breaks Suivants
0792  FD96 5C                     INCB               dans Table
0793  FD97 5C                     INCB               .
0794  FD98 C1  0A                 CMPB    #$0A      .
0795  FD9A 27  F3  FD8F           BEQ     NEXDEC-5  .
0796  FD9C AE  C5                 LDX     B,U       .
0797  FD9E AF  C6                 STX     A,U       .
0798  FDA0 6F  C5                 CLR     B,U       .
0799  FDA2 20  F0  FD94           BRA     NEXDEC    .
0800                              END
```

0000 ASSEMBLY ERRORS
0000 ASSEMBLY WARNINGS

ADE BY GESDOS 6809 MACROASSEMBLER V3.30

## RÉSUMÉ DES INSTRUCTIONS 6809

Signification des abréviations

Mnémo. : Mnémoniaur des instructions ou code assembleur
Inhér. : Code hexadécimal en mode d'adressage inhérent
Imméd. : Code hexadécimal en mode d'adressage immédiat
étendu : Code hexadécimal en mode d'adressage étendu
direct : Code hexadécimal en mode d'adressage direct avec page de base
ind. : Code hexadécimal en mode d'adressage indexé ou indirect
Op. : Opcode en hexadécimal
@ : Nombre de cycles-machine. Lorsqu'il est suivi d'un signe + , le nombre total de cycles s'obtient en ajoutant à la valeur inscrite la valeur supplémentaire contenue dans le tableau des modes indexés
# : Nombre d'octets traduisant l'encombrement mémoire. Lorsqu'il est suivi d'un signe + , le nombre total d'octets s'obtient en ajoutant à la valeur inscrite la valeur supplémentaire contenue dans le tableau des modes indexés

A, B, D, DP, CC, PC, X, Y, S, U : Registres internes du 6809
(M), (MM) : Contenu d'une mémoire 8 bits ou 16 bits
C, V, Z, N, I, H, F, E : Bits d'état du registre CC (voir paragraphe II.6)
Signe . : Bit d'état non affecté par l'opération
Signe ? : Bit d'état affecté mais résultat non significatif
Signe X : Bit d'état affecté par l'opération

| Mnémo. | Inhér. Op. @ # | Imméd. Op. @ # | Etendu Op. @ # | Direct Op. @ # | Ind. Op. @ # | Rés. d'état E F H I N Z V C |
|---|---|---|---|---|---|---|
| **Addition non signée (B) + ( X) --> (X)** | | | | | | |
| ABX | 3A 3 1 | | | | | . . . . . |
| **Addition avec retenue (A ou B) + (M) + (C) --> (A ou B)** | | | | | | |
| ADCA | | 89 2 2 | B9 5 3 | 99 4 2 | A9 4+ 2+ | . . X . X X X X |
| ADCB | | C9 2 2 | F9 5 3 | D9 4 2 | E9 4+ 2+ | . . X . X X X X |
| **Addition simple (A ou B) + (M) --> (A ou B) ; (D) + (MM) --> (D)** | | | | | | |
| ADDA | | 8B 2 2 | BB 5 3 | 9B 4 2 | AB 4+ 2+ | . . X . X X X X |
| ADDB | | CB 2 2 | FB 5 3 | DB 4 2 | EB 4+ 2+ | . . X . X X X X |
| ADDD | | C3 4 3 | F3 7 3 | D3 6 2 | E3 6+ 2+ | . . . . X X X X |
| **ET logique (A ou B ou CC) . (M) --> (A ou B ou CC)** | | | | | | |
| ANDA | | 84 2 2 | B4 5 3 | 94 4 2 | A4 4+ 2+ | . . . . X X 0 . |
| ANDB | | C4 2 2 | F4 5 3 | D4 4 2 | E4 4+ 2+ | . . . . X X 0 . |
| ANDCC | | 1C 3 2 | | | | X X X X X X X X |
| **Décalage arithmétique à gauche (C) <-- b7b6b5b4b3b2b1b0 <-- 0** | | | | | | |
| ASL | | | 78 7 3 | 08 6 2 | 68 6+ 2+ | . . ? . X X X X |
| ASLA | 48 2 1 | | | | | . . ? . X X X X |
| ASLB | 58 2 1 | | | | | . . ? . X X X X |
| **Décalage arithmétique à droite b7 --> b7b6b5b4b3b2b1b0 --> (C)** | | | | | | |
| ASR | | | 77 7 3 | 07 6 2 | 67 6+ 2+ | . . ? . X X X X |
| ASRA | 47 2 1 | | | | | . . ? . X X X X |
| ASRB | 57 2 1 | | | | | . . ? . X X X X |
| **Test de bit (A ou B) . (M) .(A ou B) non modifié, seul (CC) affecté** | | | | | | |
| BITA | | 85 2 2 | B5 5 3 | 95 4 2 | A5 4+ 2+ | . . . . X X 0 . |
| BITB | | C5 2 2 | F5 5 3 | D5 4 2 | E5 4+ 2+ | . . . . X X 0 . |

| Mnémo. | Inhér. Op. @ # | Imméd. Op. @ # | Etendu Op. @ # | Direct Op. @ # | Ind. Op. @ # | Rés. d'état E F H I N Z V C |
|---|---|---|---|---|---|---|
| **Mise à 0 (O) --> (A ou B ou M)** | | | | | | |
| CLR | | | 7F 7 3 | 0F 6 2 | 6F 6+ 2+ | . . . . 0 1 0 0 |
| CLRA | 4F 2 1 | | | | | . . . . 0 1 0 0 |
| CLRB | 5F 2 1 | | | | | . . . . 0 1 0 0 |
| **Comparaison et action sur (CC). Registres intacts. (A ou B) - (M) (D ou S ou U ou X ou Y) - (MM)** | | | | | | |
| CMPA | | B1 2 2 | B1 5 3 | 91 4 2 | A1 4+ 2+ | . . ? . X X X X |
| CMPB | | C1 2 2 | F1 5 3 | D1 4 2 | E1 4+ 2+ | . . ? . X X X X |
| CMPD | | 1083 5 4 | 10B3 8 4 | 1093 7 3 | 10A3 7+ 3+ | . . . . X X X X |
| CMPS | | 11BC 5 4 | 11BC 8 4 | 119C 7 3 | 11AC 7+ 3+ | . . . . X X X X |
| CMPU | | 1183 5 4 | 11B3 8 4 | 1193 7 3 | 11A3 7+ 3+ | . . . . X X X X |
| CMPX | | 8C 4 3 | BC 7 3 | 9C 6 2 | AC 6+ 2+ | . . . . X X X X |
| CMPY | | 108C 5 4 | 10BC 8 4 | 109C 7 3 | 10AC 7+ 3+ | . . . . X X X X |
| **Complémentation logique des registres 8 bits** | | | | | | |
| COM | | | 73 7 3 | 03 6 2 | 63 6+ 2+ | . . . . X X 0 1 |
| COMA | 43 2 1 | | | | | . . . . X X 0 1 |
| COMB | 53 2 1 | | | | | . . . . X X 0 1 |
| **Ajustement décimal de l'accumulateur A** | | | | | | |
| DAA | 19 2 1 | | | | | . . . . X X X X |
| **Décrémentation (M ou A ou B) - 1 --> (M ou A ou B)** | | | | | | |
| DEC | | | 7A 7 3 | 0A 6 2 | 6A 6+ 2+ | . . . . X X X . |
| DECA | 4A 2 1 | | | | | . . . . X X X . |
| DECB | 5A 2 1 | | | | | . . . . X X X . |
| **OU exclusif (A ou B) ⊕ (M) --> (A ou B)** | | | | | | |
| EORA | | B8 2 2 | B8 5 3 | 98 4 2 | A8 4+ 2+ | . . . . X X 0 . |
| EORB | | C8 2 2 | F8 5 3 | D8 4 2 | E8 4+ 2+ | . . . . X X 0 . |
| **Echange de deux registres** | | | | | | |
| EXG R1,R2 | 1E.. 8 2 | R1 et R2 doivent être de même type, 8 ou 16 bits. L'instruction complète s'obtient en ajoutant le post-octet au code 1E. Le post-octet est formé de 2 demi-octets dont les valeurs possibles sont: D : 0 X : 1 Y : 2 U : 3 S : 4 PC: 5 A : 8 B : 9 CC: A DP: B Exemple: EXG A,B est traduit par 1E 89 ou 1E 98 EXG X,Y est traduit par 1E 12 ou 1E 21 Aucun bit du registre d'état n'est affecté à moins que le registre echangé soit CC lui-même | | | | | |
| **Incrémentation (M ou A ou B) + 1 --> (M ou A ou B)** | | | | | | |
| INC | | | 7C 7 3 | 0C 6 2 | 6C 6+ 2+ | . . . . X X X . |
| INCA | 4C 2 1 | | | | | . . . . X X X . |
| INCB | 5C 2 1 | | | | | . . . . X X X . |
| **Chargement des registres (M) --> (A ou B) ; (MM) --> (D ou S ou U ou X ou Y)** | | | | | | |
| LDA | | B6 2 2 | B6 5 3 | 96 4 2 | A6 4+ 2+ | . . . . X X 0 . |
| LDB | | C6 2 2 | F6 5 3 | D6 4 2 | E6 4+ 2+ | . . . . X X 0 . |
| LDD | | CC 3 3 | FC 6 3 | DC 5 2 | EC 5+ 2+ | . . . . X X 0 . |
| LDS | | 10CE 4 4 | 10FE 7 4 | 10DE 6 3 | 10EE 6+ 3+ | . . . . X X 0 . |
| LDU | | CE 3 3 | FE 6 3 | DE 5 2 | EE 5+ 2+ | . . . . X X 0 . |
| LDX | | 8E 3 3 | BE 6 3 | 9E 5 2 | AE 5+ 2+ | . . . . X X 0 . |
| LDY | | 108E 4 4 | 10BE 7 4 | 109E 6 3 | 10AE 6+ 3+ | . . . . X X 0 . |

| Mnémo. | Inhér. Op. | @ | # | Imméd. Op. | @ | # | Etendu Op. | @ | # | Direct Op. | @ | # | Ind. Op. | @ | # | Res. d'état E | F | H | I | N | Z | V | C |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SBCB | | | | C2 | 2 | 2 | F2 | 5 | 3 | D2 | 4 | 2 | E2 | 4+ | 2+ | . | . | ? | . | X | X | X | X |
| Extension de signe. FF --> (A) si b7 de (B) égal à 1, 0 --> (A) si b7=0 | | | | | | | | | | | | | | | | | | | | | | | |
| SEX | 1D | 2 | 1 | | | | | | | | | | | | | . | . | . | . | X | X | . | . |
| Mise en mémoire de registres (A ou B) --> (M) | | | | | | | | | | | | | | | | | | | | | | | |
| (D ou S ou U ou X ou Y) --> (MM) | | | | | | | | | | | | | | | | | | | | | | | |
| STA | | | | | | | B7 | 5 | 3 | 97 | 4 | 2 | A7 | 4+ | 2+ | . | . | . | . | X | X | 0 | . |
| STB | | | | | | | F7 | 5 | 3 | D7 | 4 | 2 | E7 | 4+ | 2+ | . | . | . | . | X | X | 0 | . |
| STD | | | | | | | FD | 6 | 3 | DD | 5 | 2 | ED | 5+ | 2+ | . | . | . | . | X | X | 0 | . |
| STS | | | | | | | 10FF | 7 | 4 | 10DF | 6 | 3 | 10EF | 6+ | 3+ | . | . | . | . | X | X | 0 | . |
| STU | | | | | | | FF | 6 | 3 | DF | 5 | 2 | EF | 5+ | 2+ | . | . | . | . | X | X | 0 | . |
| STX | | | | | | | BF | 6 | 3 | 9F | 5 | 2 | AF | 5+ | 2+ | . | . | . | . | X | X | 0 | . |
| STY | | | | | | | 10BF | 7 | 4 | 109F | 6 | 3 | 10AF | 6+ | 3+ | . | . | . | . | X | X | 0 | . |
| Soustraction (A ou B) - (M) --> (A ou B) ; (D) - (MM) --> (D) | | | | | | | | | | | | | | | | | | | | | | | |
| SUBA | | | | B0 | 2 | 2 | B0 | 5 | 3 | 90 | 4 | 2 | A0 | 4+ | 2+ | . | . | ? | . | X | X | X | X |
| SUBB | | | | C0 | 2 | 2 | F0 | 5 | 3 | D0 | 4 | 2 | E0 | 4+ | 2+ | . | . | ? | . | X | X | X | X |
| SUBD | | | | 83 | 4 | 3 | B3 | 7 | 3 | 93 | 6 | 2 | A3 | 6+ | 2+ | . | . | . | . | X | X | X | X |
| Transfert de registres B et 16 bits | | | | | | | | | | | | | | | | | | | | | | | |
| TFR R1,R2 | 1F.. | 7 | 2 | | | | | | | | | | | | | voir commentaire | | | | | | | |
| Les registres R1 et R2 doivent être de même longueur. Le registre d'état n'est affecté que si R2 est CC lui-même. Le calcul du post-octet est identique à celui exposé pour l'instruction EXG R1,R2 | | | | | | | | | | | | | | | | | | | | | | | |
| Test et action sur (CC) (M ou A ou B) - (0) | | | | | | | | | | | | | | | | | | | | | | | |
| TST | | | | | | | 7D | 7 | 3 | 0D | 6 | 2 | 6D | 6+ | 2+ | . | . | . | . | X | X | 0 | . |
| TSTA | 4D | 2 | 1 | | | | | | | | | | | | | . | . | . | . | X | X | 0 | . |
| TSTB | 5D | 2 | 1 | | | | | | | | | | | | | . | . | . | . | X | X | 0 | . |

INSTRUCTIONS IMPLIQUANT UNE RUPTURE DE SEQUENCE

| Mnémo. | Inhér. Op. | @ | # | Imméd. Op. | @ | # | Etendu Op. | @ | # | Direct Op. | @ | # | Ind. Op. | @ | # | Res. d'état E | F | H | I | N | Z | V | C |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Saut inconditionnel vers une adresse | | | | | | | | | | | | | | | | | | | | | | | |
| JMP | | | | | | | 7E | 4 | 3 | 0E | 3 | 2 | 6E | 3+ | 2+ | . | . | . | . | . | . | . | . |
| Saut vers un sous-programme avec sauvegarde de l'adresse de retour | | | | | | | | | | | | | | | | | | | | | | | |
| JSR | | | | | | | BD | 8 | 3 | 9D | 7 | 2 | AD | 7+ | 2+ | . | . | . | . | . | . | . | . |
| Retour de sous-programme d'interruption | | | | | | | | | | | | | | | | | | | | | | | |
| RTI | 3B | 6/15 | 1 | | | | | | | | | | | | | X | X | X | X | X | X | X | X |
| Si E=0, le nombre de cycles est 6. Si E=1, le nombre de cycles est 15 | | | | | | | | | | | | | | | | | | | | | | | |
| Retour de sous-programme | | | | | | | | | | | | | | | | | | | | | | | |
| RTS | 39 | 5 | 1 | | | | | | | | | | | | | . | . | . | . | . | . | . | . |
| Interruptions logicielles | | | | | | | | | | | | | | | | | | | | | | | |
| SWI | 3F | 19 | 1 | | | | | | | | | | | | | 1 | 1 | . | 1 | . | . | . | . |
| SWI2 | 103F | 20 | 2 | | | | | | | | | | | | | 1 | . | . | . | . | . | . | . |
| SWI3 | 113F | 20 | 2 | | | | | | | | | | | | | 1 | . | . | . | . | . | . | . |
| Syncronisation avec la ligne d'interruption | | | | | | | | | | | | | | | | | | | | | | | |
| SYNC | 13 | 2 | 1 | | | | | | | | | | | | | . | . | . | . | . | . | . | . |
| ET logique de (CC) avec un opérande immédiat puis attente d'interruption | | | | | | | | | | | | | | | | | | | | | | | |
| CWAI | 3C | 20 | 2 | | | | | | | | | | | | | X | X | X | X | X | X | X | X |

| Mnémo. | Inhér. Op. | @ | # | Imméd. Op. | @ | # | Etendu Op. | @ | # | Direct Op. | @ | # | Ind. Op. | @ | # | Res. d'état E | F | H | I | N | Z |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Chargement de l'adresse effective | | | | | | | | | | | | | | | | | | | | | |
| LEAS | | | | | | | | | | | | | 32 | 4+ | 2+ | . | . | . | . | . | . |
| LEAU | | | | | | | | | | | | | 33 | 4+ | 2+ | . | . | . | . | . | . |
| LEAX | | | | | | | | | | | | | 30 | 4+ | 2+ | . | . | . | . | . | X |
| LEAY | | | | | | | | | | | | | 31 | 4+ | 2+ | . | . | . | . | . | X |
| LSL, LSLA, LSLB sont identiques à ASL, ASLA, ASLB | | | | | | | | | | | | | | | | | | | | | |
| Décalage logique à droite 0 --> b7b6b5b4b3b2b1b0 --> (C) | | | | | | | | | | | | | | | | | | | | | |
| LSR | | | | | | | | | | 74 | 7 | 3 | 04 | 6 | 2 | 64 | 6+ | 2+ | . | . | . |
| LSRA | 44 | 2 | 1 | | | | | | | | | | | | | | | | . | . | . |
| LSRB | 54 | 2 | 1 | | | | | | | | | | | | | | | | . | . | . |
| Multiplication non signée (A) * (B) --> (D) | | | | | | | | | | | | | | | | | | | | | |
| MUL | 3D | 11 | 1 | | | | | | | | | | | | | | | | | | |
| Complémentation à 2 (0) - (M ou A ou B) --> (M ou A ou B) | | | | | | | | | | | | | | | | | | | | | |
| NEG | | | | | | | | | | 70 | 7 | 3 | 00 | 6 | 2 | 60 | 6+ | 2+ | . | . | ? |
| NEGA | 40 | 2 | 1 | | | | | | | | | | | | | | | | . | . | ? |
| NEGB | 50 | 2 | 1 | | | | | | | | | | | | | | | | . | . | ? |
| Sans opération | | | | | | | | | | | | | | | | | | | | | |
| NOP | 12 | 2 | 1 | | | | | | | | | | | | | | | | | | |
| OU logique (A ou B ou CC) + (M) --> (A ou B ou CC) | | | | | | | | | | | | | | | | | | | | | |
| ORA | | | | | | | BA | 2 | 2 | BA | 5 | 3 | 9A | 4 | 2 | AA | 4+ | 2+ | . | . | . |
| ORB | | | | | | | CA | 2 | 2 | FA | 5 | 3 | DA | 4 | 2 | EA | 4+ | 2+ | . | . | . |
| ORCC | | | | | | | 1A | 3 | 2 | | | | | | | | | | X | X | X |
| Instructions d'empilement ou de dépilement | | | | | | | | | | | | | | | | | | | | | |
| PSHS | 34.. | 5+ | 2 | | | | | | | | | | | | | 7 6 5 4 3 2 1 0 — No. du bit | | | | | |
| PSHU | 36.. | 5+ | 2 | | | | | | | | | | | | | | | | | | |
| PULS | 35.. | 5+ | 2 | | | | | | | | | | | | | PC U/S Y X DP B A CC — Res. | | | | | |
| PULU | 37.. | 5+ | 2 | | | | | | | | | | | | | | | | | | |

Pour trouver le post-octet correspondant dans une instruction d'empilement ou de dépilement, inscrire un 1 dans la case correspondante, puis calculer la valeur globale de l'octet. Exemples:
PSHS U,X,A,CC a pour code 34 53
PSHS A a pour code 34 02
Pour PULS et PULU, le registre d'état n'est pas affecté à moins que le dépilement invoque lui-même CC
Pour connaitre le nombre total de cycles requis, ajouter à la valeur de base 5, 1 cycle pour chaque registre B ou CC et 2 cycles pour chaque registre 16 bits

| Mnémo. | Inhér. Op. | @ | # | Imméd. Op. | @ | # | Etendu Op. | @ | # | Direct Op. | @ | # | Ind. Op. | @ | # | Res. d'état E | F | H | I | N | Z |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Décalage circulaire à gauche (C) <-- b7b6b5b4b3b2b1b0 <-- (C) | | | | | | | | | | | | | | | | | | | | | |
| ROL | | | | | | | | | | 79 | 7 | 3 | 09 | 6 | 2 | 69 | 6+ | 2+ | . | . | . |
| ROLA | 49 | 2 | 1 | | | | | | | | | | | | | | | | . | . | . |
| ROLB | 59 | 2 | 1 | | | | | | | | | | | | | | | | . | . | . |
| Décalage circulaire à droite (C) --> b7b6b5b4b3b2b1b0 --> (C) | | | | | | | | | | | | | | | | | | | | | |
| ROR | | | | | | | | | | 76 | 7 | 3 | 06 | 6 | 2 | 66 | 6+ | 2+ | . | . | . |
| RORA | 46 | 2 | 1 | | | | | | | | | | | | | | | | . | . | . |
| RORB | 56 | 2 | 1 | | | | | | | | | | | | | | | | . | . | . |
| Soustraction avec retenue (A ou B) - (M) - (C) --> (A ou B) | | | | | | | | | | | | | | | | | | | | | |
| SBCA | | | | | | | B2 | 2 | 2 | B2 | 5 | 3 | 92 | 4 | 2 | A2 | 4+ | 2+ | . | . | ? |

NSTRUCTIONS DE BRANCHEMENTS RELATIFS COURTS ET LONGS. Le mode d'adressage
orrespondant est le mode relatif.

| émo. | Op. | @ | # | Mnémo. | Op. | @ | # | Conditions |
|---|---|---|---|---|---|---|---|---|
| CC | 24 | 3 | 2 | LBCC | 1024 | 5/6 | 4 | C = 0 |
| CS | 25 | 3 | 2 | LBCS | 1025 | 5/6 | 4 | C = 1 |
| EQ | 27 | 3 | 2 | LBEQ | 1027 | 5/6 | 4 | Z = 1 |
| GE | 2C | 3 | 2 | LBGE | 102C | 5/6 | 4 | N ⊕ V = 0 |
| GT | 2E | 3 | 2 | LBGT | 102E | 5/6 | 4 | Z + (N ⊕ V) = 0 |
| HI | 22 | 3 | 2 | LBHI | 1022 | 5/6 | 4 | Z + C = 0 |
| HS | 24 | 3 | 2 | LBHS | 1024 | 5/6 | 4 | C = 0 |
| LE | 2F | 3 | 2 | LBLE | 102F | 5/6 | 4 | (N ⊕ V) + Z = 1 |
| LO | 25 | 3 | 2 | LBLO | 1025 | 5/6 | 4 | C = 1 |
| LS | 23 | 3 | 2 | LBLS | 1023 | 5/6 | 4 | C + Z 1 |
| LT | 2D | 3 | 2 | LBLT | 102D | 5/6 | 4 | N ⊕ V = 1 |
| MI | 2B | 3 | 2 | LBMI | 102B | 5/6 | 4 | N = 1 |
| NE | 26 | 3 | 2 | LBNE | 1026 | 5/6 | 4 | Z = 0 |
| PL | 2A | 3 | 2 | LBPL | 102A | 5/6 | 4 | N = 0 |
| RA | 20 | 3 | 2 | LBRA | 16 | 5 | 3 | Aucune |
| RN | 21 | 3 | 2 | LBRN | 1021 | 5 | 4 | voir note |
| SR | 8D | 7 | 2 | LBSR | 17 | 9 | 3 | Aucune |
| VC | 28 | 3 | 2 | LBVC | 1028 | 5/6 | 4 | V = 0 |
| VS | 29 | 3 | 2 | LBVS | 1029 | 5/6 | 4 | V = 1 |

te: 5/6 signifie 6 cycles quand le branchement s'opère, 5 cycles quand la con-
dition de branchement n'est pas realisée.
BRN équivaut à 2 NOP. LBRN équivaut à 4 NOP

ADRESSAGE INDEXE. SIGNIFICATION DES BITS DU POST-OCTET.

| Bit du registre post-octet 7 6 5 4 | 3 2 1 0 | Mode d'adressage indexé |
|---|---|---|
| 0 r r S. | X X X X | R + 4 bits déplacement |
| 1 r r 0 | 0 0 0 0 | ,R+ |
| 1 r r I. | 0 0 0 1 | ,R++ |
| 1 r r 0 | 0 0 1 0 | ,-R |
| 1 r r I. | 0 0 1 1 | ,--R |
| 1 r r I. | 0 1 0 0 | R + 0 déplacement |
| 1 r r I. | 0 1 0 1 | R + Acc.B déplacement |
| 1 r r I. | 0 1 1 0 | R + Acc.A déplacement |
| 1 r r I. | 1 0 0 0 | R + 7 bits déplacement |
| 1 r r I. | 1 0 0 1 | R + 16 bits déplacement |
| 1 r r I. | 1 0 1 1 | R + Acc.D déplacement |
| 1 X X I. | 1 1 0 0 | PC + 7 bits déplacement |
| 1 X X I. | 1 1 0 1 | PC + 15 bits déplacement |
| 1 X X 1 | 1 1 1 1 | Mode indirect type [n] |

R: Registre base

X: indiffér

→ Champ définissant le mode d'adressage

→ Bit d'indirection. Si I.=1, mode indirect
Dans le 1er cas où b7=0, ce bit devient le
de signe pour l'offset. Si S.=1, l'offset
sur 4 bits se retranche de l'adresse de ba

bit codé se la

→ Les 2 bits b6b5 du post-octet définissent
nature du registre de base du mode indexé.
Si b6b5 = 00 le registre est X
Si b6b5 = 01 le registre est Y
Si b6b5 = 10 le registre est U
Si b6b5 = 11 le registre est S

NOMBRE DE CYCLES ET D'OCTETS ADDITIONNELS POUR LES MODES INDEXES OU INDIRE
CTS

| Formes | Non indirect Assembleur | Post-octet | @ | # | Indirect Assembleur | Post-octet |
|---|---|---|---|---|---|---|
| sans deplacement | ,R | 1rr0 0100 | 0 | 0 | [,R] | 1rr1 0100 |
| deplac. 4 bits | n,R | 0rrS XXXX | 1 | 0 | par défaut, mode 7 bits | |
| deplac. 7 bits | n,R | 1rr0 1000 | 1 | 1 | [n,R] | 1rr1 1000 |
| deplac. 15 bits | n,R | 1rr0 1001 | 4 | 2 | [n,R] | 1rr1 1001 |
| deplac. Acc.A | A,R | 1rr0 0110 | 1 | 0 | [A,R] | 1rr1 0110 |
| deplac. Acc.B | B,R | 1rr0 0101 | 1 | 0 | [B,R] | 1rr1 0101 |
| deplac. Acc.D | D,R | 1rr0 1011 | 4 | 0 | [D,R] | 1rr1 1011 |
| Increm. par 1 | ,R+ | 1rr0 0000 | 2 | 0 | impossible | |
| Increm. par 2 | ,R++ | 1rr0 0001 | 3 | 0 | [,R++] | 1rr1 0001 |
| decrem. par 1 | ,-R | 1rr0 0010 | 2 | 0 | impossible | |
| decrem. par 2 | ,--R | 1rr0 0011 | 3 | 0 | [,--R] | 1rr1 0011 |
| depl. 7 bits PCR | n,PCR | 1XX0 1100 | 1 | 1 | [n,PCR] | 1XX1 1100 |
| depl.15 bits PCR | n,PCR | 1XX0 1101 | 5 | 2 | [n,PCR] | 1XX1 1101 |
| Indirect étendu | — | — | — | | [n] | 1001 1111 |

# BIBLIOGRAPHIE.


— PROGRAMMATION en ASSEMBLEUR 6809.

par BUI MINH DUC (Ed. EYROLLES)

— DOCUMENTS MOTOROLA.