Réseaux
et systèmes
d'information

Bull

MANUEL D'UTILISATION

SOLAR

IOCS

Moniteur de gestion des entrées-sorties

LOGICIEL

LOGICIEL

LOGICIEL

Bull

1 164 151 02 030

Réseaux
et systèmes
d'information

Bull

MANUEL D'UTILISATION

I O C S

Gamme SOLAR

Systèmes :

Objet :

Date d'édition FEVRIER 1984

Référence document : 1 164 151 02 030 13 / FR

SOMMAIRE

PREMIERE PARTIE :

IOCS

1 - DESCRIPTION D'IOCS

1.1 - DESCRIPTION SOMMAIRE DU FONCTIONNEMENT

Du point de vue fonctionnel, IOCS comporte deux types de modules.:

a) Un module se déroulant sous niveau software de la tache appelante.

Ce module est activé par le superviseur chaque fois qu'une tache fait une SVC IOCS.

Son rôle est le suivant :

- **il analyse la demande en vérifiant l'IOCB**
- **il attribue au demandeur les ressources nécessaires à l'échange (unité physique, buffer du "pool")**
- **il active le module du driver chargé d'initialiser les échanges sur le périphérique concerné**
- **il analyse le compte-rendu du driver après qu'il ait initialisé les échanges.**
- **suivant le type de demande rend le contrôle à l'appelant ou le met en attente d'événement fin d'échange.**

b) Des tâches hardware de niveaux déterminés à la configuration du système.

Ces tâches prennent en compte les interruptions en provenance des périphériques.

Leur rôle est le suivant :

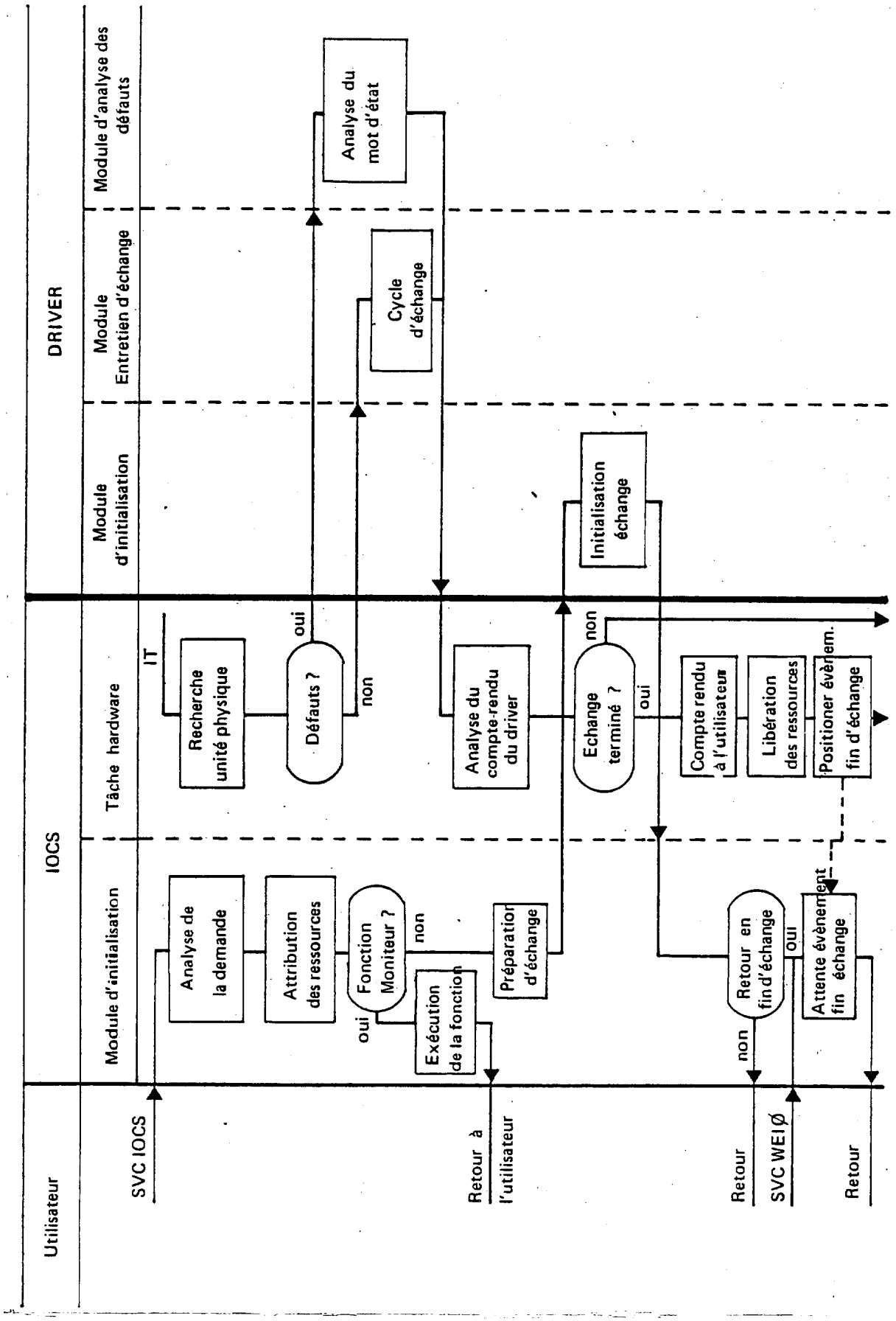
- **reconnaître l'origine de l'interruption (le périphérique qui a provoqué l'interruption)**
- **reconnaître la nature de l'interruption (normale ou exception)**
- **activer le module du driver chargé d'analyser la nature de l'interruption**
- **en fin d'échange libérer les ressources qui ont été attribuées lors de l'initialisation de l'échange**
- **positionner l'événement "fin d'échange".**

1.2 - GESTION DES RESSOURCES ET EVENEMENTS FIN D'ECHANGE

1.2.1 - IOCS versions multi-tâches (IOCS - M1/M2)

IOCS gère des demandes en provenance de plusieurs tâches software portant, éventuellement, sur les mêmes périphériques.

Lorsque plusieurs tâches requièrent une même ressource (périphérique, buffer du "pool", contexte canal) elles sont mises en attente dans une file suivant leurs priorités. En général, ces files sont réalisées à l'aide d'un sémaphore d'exclusion.



DÉROULEMENT D'UN ÉCHANGE

a) Ressource unité physique

Chaque unité physique est considérée comme une ressource gérée par un sémaphore d'exclusion. A chaque unité physique est affecté un sémaphore d'accès à cette ressource (SEMGEN sémaphore à un seul accès). Une ressource unité physique est attribuée à une tâche :

- soit pour le déroulement d'un échange. Dans ce cas l'accès à la ressource unité physique est libéré en fin d'échange
- soit parce que celle-ci a demandé l'attachement (fonction PUA).
  Dans ce cas la ressource sera libérée sur la fonction détachement (PUD) si l'unité physique était attachée.

b) Ressource "pool buffer"

L'accès au pool est géré par un sémaphore d'exclusion (SEMBUF) dont le nombre d'accès est égal au nombre de buffers du pool.

Lorsqu'une tâche fait une demande d'échange avec utilisation du "pool buffer", on lui attribue un accès à cette ressource. Cet accès sera libéré en fin d'échange.

c) Ressource "contexte canal"

Les canaux HDC ne peuvent gérer simultanément que deux échanges. L'accès à un tel canal est donc filtré par un sémaphore d'exclusion à deux accès. L'attribution de la ressource "contexte canal" n'est faite que si le périphérique est connecté à un canal du type HDC.

d) "Fin d'échange"

A chaque demande d'échange (exceptées les demandes avec utilisation du pool buffer), IOCS associe un sémaphore privé réalisé dans le mot 4 de l'IOCB correspondant à cet échange.

A l'initialisation de l'échange, le compteur d'accès à ce sémaphore est nul.

L'accès à ce sémaphore est fourni par la fin d'échange (événement fin d'échange).

Lors d'une demande avec retour en fin d'échange, IOCS demande l'accès à ce semaphore avant de rendre le contrôle à l'utilisateur. Si la fin d'échange n'est pas encorearrivée la tâche est suspendue et réactivée par l'événement fin d'échange.

Il en est de même lorsqu'une tâche se met en attente de fin d'échange sur une SVC WEIO.

- on associe en outre à chaque unité physique un sémaphore privé utilisé pour synchroniser les demandes d'échange et les fins d'échange portant sur cette unité physique lorsque celle-ci est attachée à une tache (sémaphore SEMATT).

1.2.2 - IOCS version mono-tâche (IOCS-S)

Cette version d'IOCS peut gérer les échanges provenant d'une seule tache.
La gestion de ses demandes en est donc extrêmement simplifiée.

a) Ressource unité physique

On considérera qu'une unité physique est "occupée logiquement" lorsqu'elle est en cours d'échange. Cette occupation logique est mémorisée par un indicateur (BITENC).

Lorsqu'une tâche demande un échange sur une unité physique, elle est mise en attente tant que l'unité physique en question est en cours d'échange.

b) Evénement fin d'échange

Pour des raisons de compatibilité avec l'autre version d'IOCS, on utilise le mot (4) de l'IOCB pour mémoriser si l'échange correspondant à cet IOCB est termine ou non. C'est le mot (4) de l'IOCB qui est donc testé lorsqu'une tâche se met en attente de fin d'échange par un SVC WEIO.

Le déroulement d'une tâche faisant des demandes à IOCS est donc rigoureusement identique lorsqu'elle travaille seule en utilisant IOCS "mono-tâche" ou lorsqu'elle travaille intégrée à un système multi-taches utilisant IOCS "multi-tâches".

1.3 - LES TABLES SYSTEME

Pour communiquer entre eux, les différents modules d'IOCS et les drivers utilisent des tables (Tables Système). Ces tables sont décrites dans le présent paragraphe.

1.3.1 - Tables d'unités symboliques : TBS

Cette table est organisée en octets. Elle est pointée par les numéros d'unités symboliques. Elle fait correspondre à chaque numéro d'unité symbolique un numéro d'unité fonctionnelle.
Elle peut être modifiée par le programme superviseur (BOS par exemple) lors des affectations d'unités fonctionnelles aux unités symboliques.

TBS

N° FU associée à la SUO	N° FU associée à la SU 1
N° FU associée à la SU2	
N° FU associée à la SUi	

La taille de cette table (nombre d'octets) est contenue dans la mémoire NUSMAX.
Les unités symboliques sont numérotées à partir de 0.

1.3.2 - Table d'unités fonctionnelles : TBF

Cette table est organisée en mots. Elle est pointée par les numéros d'unités fonctionnelles et fait correspondre à chaque numéro d'unité fonctionnelle l'adresse de la table unité physique sur laquelle elle pointe (voir 1.3.4).

TBF

Adresse TUP correspondant à la FU1
Adresse TUP correspondant à la FU2
Adresse TUP correspondant à la Fui

La taille de cette table est contenue dans la mémoire NUFMAX. Les unités fonctionnelles sont numérotées à partir de 1.
Cette table est initialisée à la génération.

1.3.3 - Table des niveaux d'interruption des unités fonctionnelles : TBNFU

Cette table est organisée en octets. Elle est pointée par les numéros d'unités fonctionnelles et fait correspondre à chaque numéro d'unité fonctionnelle le numéro d'interruption correspondant.

TBNFU

N° IT correspondant à la FU = 00	N° IT correspondant à la FU = 01
N° IT correspondant à la FU = 04	
- - - - - - - - - - - - - - - -	
- - - - - - - - - - - - - - - -	

La taille de cette table (en mot) est égale à (NUFMAX+ 1) /2. Elle est initialisée à la génération.

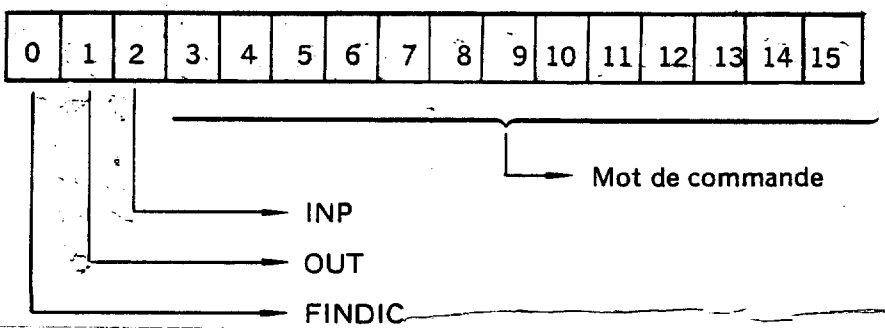
1.3.4 - Table des mots de commande : TBMCOM

Cette table est organisée en mots. Elle est pointée par les numéros d'unités fonctionnelles et fait correspondre à chaque numéro d'unité fonctionnelle d'une part un mot de commande d'autre part un certain nombre d'indicateurs caractérisant cette unité fonctionnelle.

TBMCOM

Mot de commande associé à la FU1
Mot de commande associé à la FU2
Mot de commande associé à la FUi

Forme d'un élément de la table :



Signification des indicateurs :

INP (bit 2) : 0 les entrées sont autorisées sur l'unité fonctionnelle
1 les entrées sont refusées sur l'unité fonctionnelle

OUT (bit1) : 0 les sorties sont autorisées sur l'unité fonctionnelle
1 les sorties sont refusées sur l'unité fonctionnelle

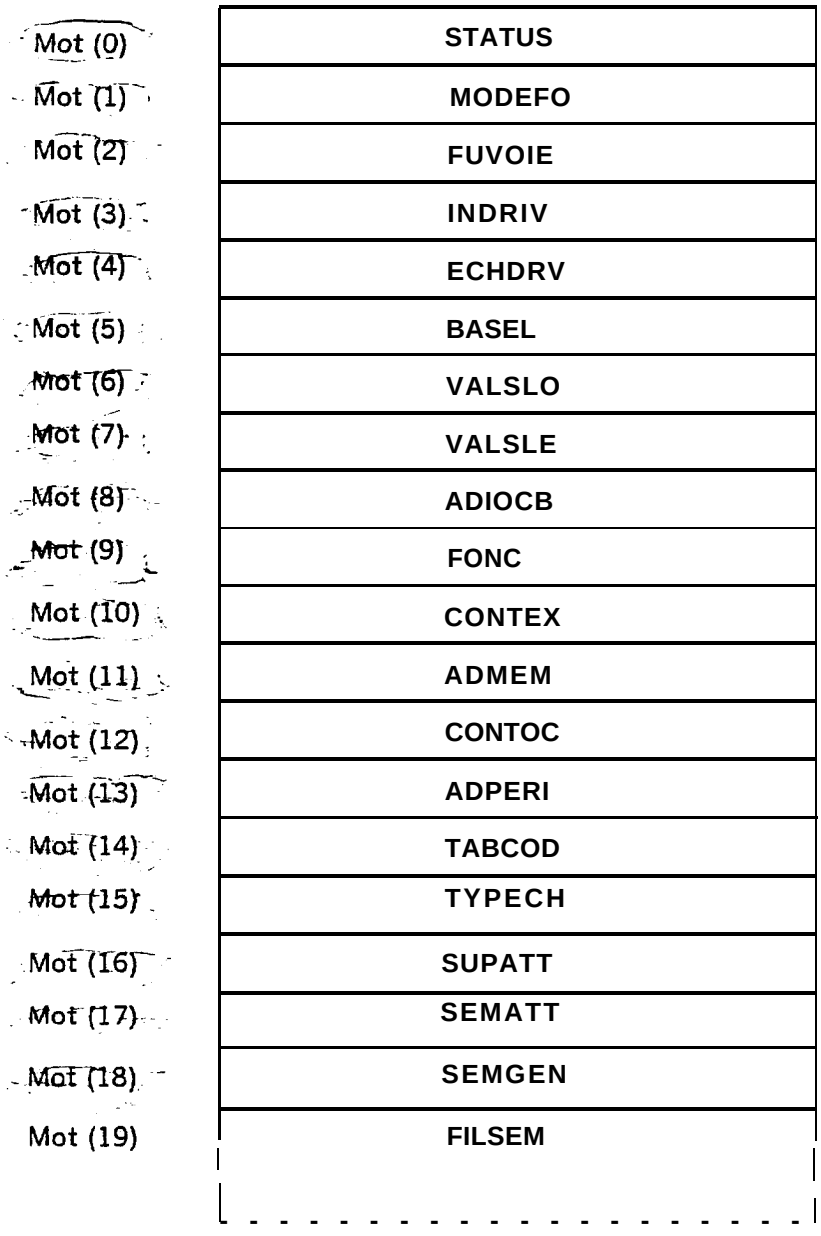
Ces deux indicateurs permettent à IOCS de s'assurer que le sens de l'échange est possible sur le périphérique concerné.

FINDIC (bit 0) : Indicateur servant éventuellement au driver pour distinguer une unité fonctionnelle particulière lorsqu'il est utilisé par plusieurs unités fonctionnelles. Il s'agit d'une convention avec le driver. Par exemple ce bit sera à 1 pour l'unité fonctionnelle TP, ce qui permettra au driver télétype de savoir qu'il doit envoyer les codes "perfo on" et "perfo off" en début et fin d'échange.

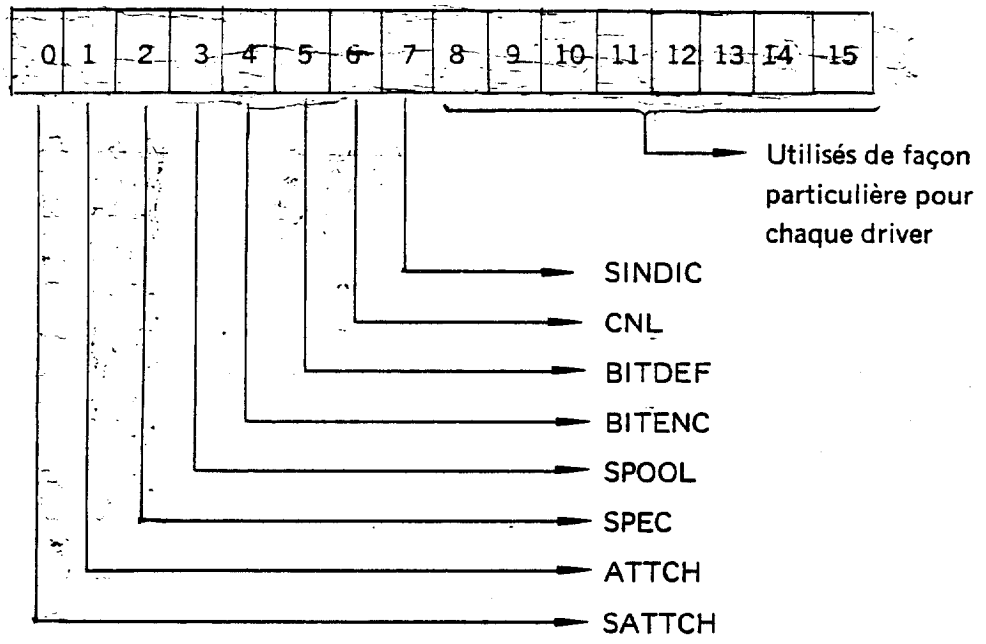
La taille de cette table (nombre de mots) est contenue dans la mémoire NUFMAX..
La table est initialisée à la configuration d'IOCS.

1.3.5 - Table unités physiques. TUP xxx

A chaque unité physique est associée une table d'unité physique TUP xxx qui contient toutes les informations (statiques et dynamiques) propres à une unité physique pour un échange donné. C'est là que le driver trouvera toutes les données propres à un échange.



Mot (0) : STATUS



Signification des différents indicateurs lorsqu'ils sont à 1

SATTCH : **L'unité physique est super-attachée. Ce bit n'est modifié que par une fonction spéciale adressée au moniteur.**

ATTCH : **L'unité physique est attachée (bit modifié par une fonction spéciale adressée au moniteur)**

SPEC : **Ce bit est réservé pour une fonction spéciale non standard que l'utilisateur voudrait inclure à son système.**

SPOOL : **L'échange demandé utilise le pool de buffers. Ce bit est positionné à 1 ou 0 à chaque demande à IOCS.**

BITENC : **Un échange est en cours sur l'unité physique**

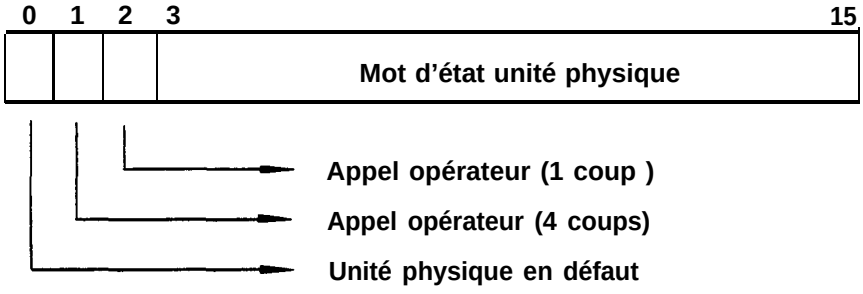
BITDEF : **Un défaut a eu lieu en cours d'échange, obligeant à abandonner l'échange. Ce bit est positionné à 1 par le module d'analyse des défauts du driver.**

CNL : **L'unité physique fonctionne en mode canal (bit positionné à la configuration).**

SINDIC : **Indicateur caractérisant l'unité fonctionnelle en échange sur l'unité physique. C'est en fait le bit FINDIC qui est recopié par IOCS à chaque demande d'échange ou de positionnement.**
Par exemple : le driver télétype trouvera ce bit à 1 lorsqu'il travaillera pour l'unité fonctionnelle TP ce qui lui permettra de savoir qu'il doit envoyer les codes "perfo on" et "perfo off" en début et fin d'échange. Chaque driver pourra l'utiliser de façon personnelle.
L'octet de droite est utilisé de façon particulière par chaque driver pour mémoriser des informations

Les bits 8 à 15 sont utilisés de façon particulière par chaque driver.
Cependant, ceux des périphériques gérant l'appel opérateur utilisent les bits 13, 14, 15 pour comptabiliser le nombre d'appels. Lors de la prise en compte d'un appel, les sous-programmes superviseurs (TAPS, RMDEF) doivent mettre à zéro ces trois bits.

Mot (1) : MODEFO



IOCS range les mots d'états des unités physiques que lui fournissent les modules d'analyse des défauts des drivers. Ces mots sont scrutés par les différents moniteur (BOS par exemple) qui peuvent alors signaler par un message d'erreur les défauts périphériques qui se sont produits.

Mot (2) : FUVOIE

Octet de gauche : numéro de FU qui est en cours d'échange
Octet de droite : numéro de voie (pour les périphériques multiplexés uniquement).

Mot (3) : INDRIV

Adresse du driver (chargée à la configuration)

Mot (4) : ECHDRV
Adresse à lancer à la prochaine interruption du périphérique (adresse d'entretien d'échange)

Mot (5) : BASEL

Il s'agit de l'adresse sur laquelle pointe la base L dans le local du driver. Cette adresse est chargée dans le mot (5) à la génération d'IOCS.
IOCS charge la base L par cette valeur chaque fois qu'il fait appel à un module du driver. Les modules des drivers n'auront donc pas à se préoccuper de charger les bases.

Mot (6) : VALSLO

Valeur du registre SLO de l'appelant qui a initialisé l'échange.

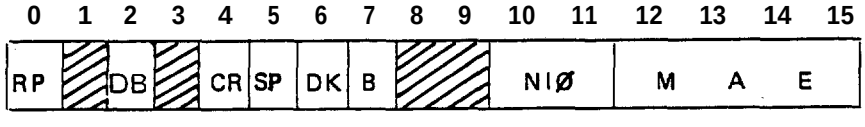
Mot (7) : VALSLE

Valeur du registre SLE de l'appelant qui a initialisé l'échange.

Mot (8) : ADIOCB

Adresse de l'IOCB pour l'échange considéré.

Mot (9) : FONC



RP : 0 Ce bit est toujours à 0

DB : 1 Mode DEBUG (Positionné par GENIO)

CR : 1 La fonction demandée au canal est un compte-rendu
CR : 0 La fonction demandée au canal est une initialisation d'échange

SP : 1 Micro programmation spécifique
SP : 0 Micro programmation standard

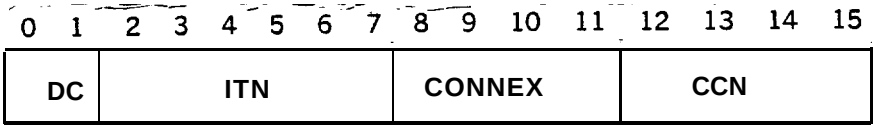
DK : 1 Le périphérique est un disque à têtes mobiles
DK : 0 Le périphérique n'est pas un disque

B : 0 Canal à mots
B : 1 Canal à octets ou fausse lecture disque (si DK = 1)

NIO : Numéro de processeur d'entrées-sorties qui gère l'échange

MAE : Extension d'adresse buffer (poids forts)

Mot (10) : CONTEX



DC : 00 le canal est de type LDC
DC : 11 le canal est de type MDC
DC : 10 le canal est de type HDC

ITN : Niveau d'interruption normale

CONNEX : Type de connexion du périphérique

0000	**coupleur standard**
0001	**coupleur asynchrone 1 voie**
0010	**coupleur multiplexé 4 voies**
0011	**coupleur multiplexé 16 voies**
0100	**coupleur multiplexé 8 voies**

CCN : Numéro de contexte hardware

Mot (11) : ADMEM

Adresse mémoire du buffer d'entrées-sorties (poids faibles)

Mot (12) : CONTOC

Compte de mots ou d'octets (suivant le coupleur) à échanger.

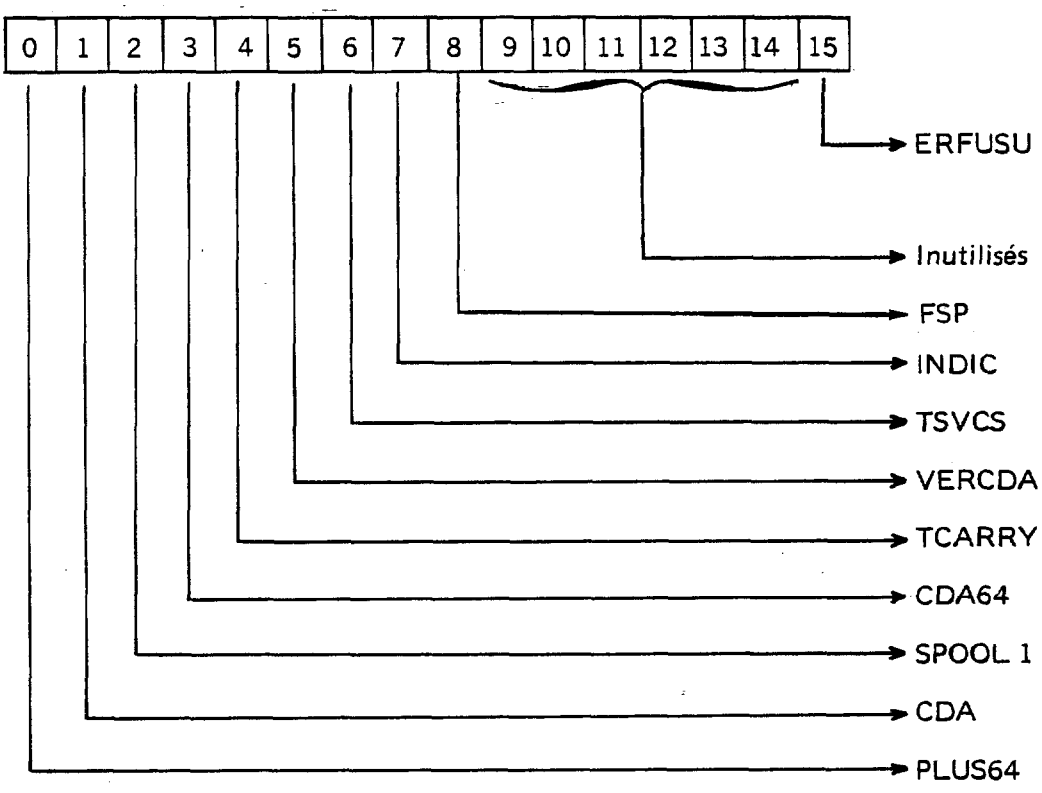
Mot (13) : ADPERI

Adresse du périphérique

Mot (14) : TABCOD

Adresse mémoire de la table des codes d'arrêt ou adresse disque (si DK = 1)

Mot (15) : TYPECH



Signification des différents indicateurs lorsqu'ils sont à 1 :

PLUS64 : Echange avec interface système
CDA : Echange avec buffer en CDA
SPOOL1 : Echange avec utilisation du pool buffer
CDA64 : Echange avec interface système ou buffer en CDA
TCARRY : Appelant en mode maître
VERCDA : Vérification d'une adresse en CDA
TSVCS : Valeur du bit SVCS du registre STATUS à l'initialisation
INDIC : Type d'adressage
***0 = adressage d'un IOCB**
***1 = adressage d'un buffer**
FSP : L'échange en cours est une fonction spéciale
ERFUSU : Unité d'échange non gérée par IOCS.

Mot (16) : SUPPAT
Sémaphore privé de super attachement.

Mot (17) : SEMATT

Sémaphore privé d'attachement.

Mot (18) : SEMGEN

Sémaphore d'exclusion gérant la ressource unité physique.

Mot (19) : FILSEM

Début de la file du sémaphore d'exclusion. Le nombre de mots réservés à la file du sémaphore d'exclusion dépend du nombre des tâches du système auquel il sera inséré (maximum 8 mots c'est-à-dire 128 taches).

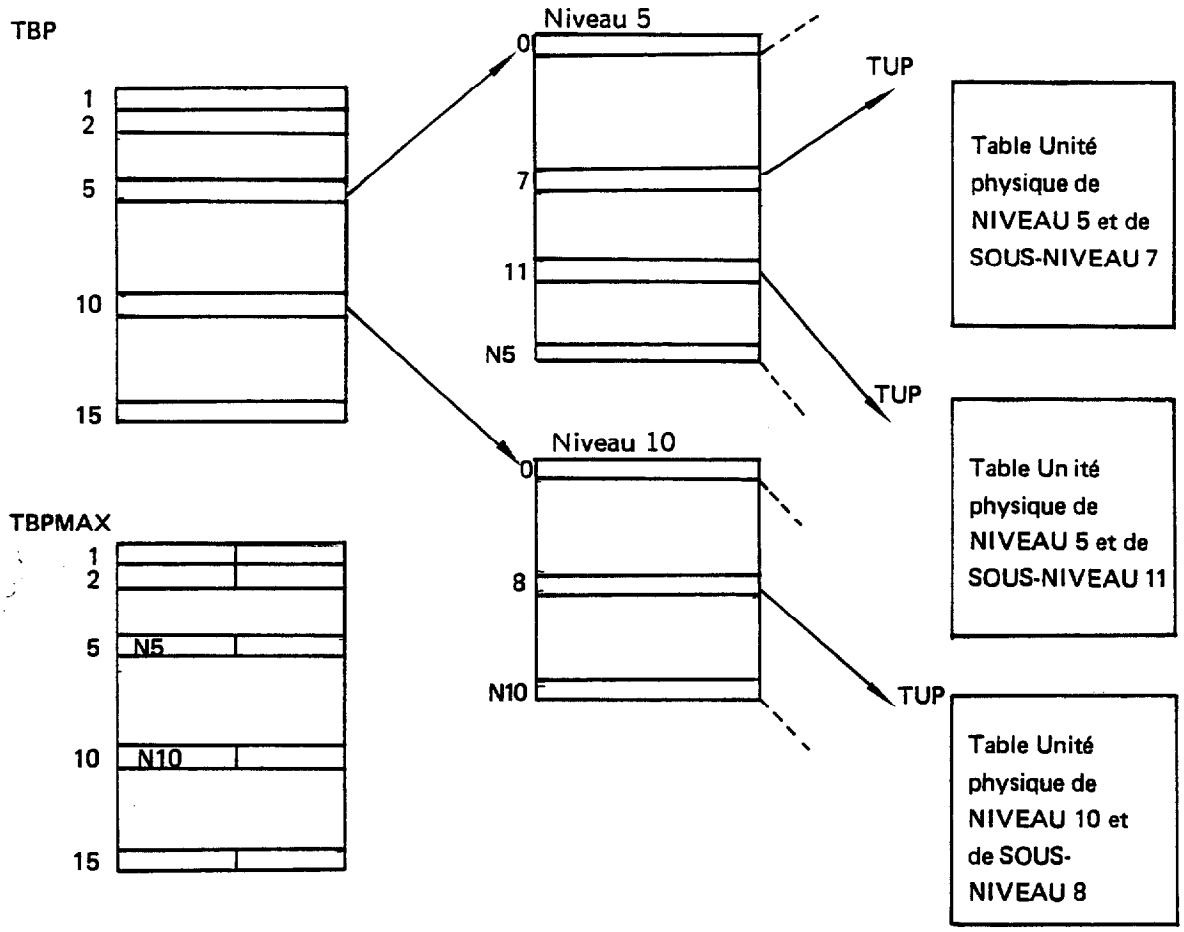
Remarque :
La table unité physique que l'on vient de décrire est la forme la plus simple de la table unité physique.
Certains drivers ont besoin d'une zone de travail supplémentaire, la table unité physique la plus générale aura la forme suivante :

1.3.6 - Table des adresses de tables unités physiques : TBP

L'accès aux tables d'unités physiques est fait en deux étapes :

a) En fonction du niveau d'interruption traité on recherche, dans la TBP, l'adresse de la table des unités physiques de ce niveau.

b) En fonction du sous-niveau appelant on a l'adresse de la table d'unité physique.



L'octet gauche de la table TBPMAX indique le rang maximum du sous-niveau géré dans chacun des niveaux gérés par IOCS.

Dans le cas où l'on a plusieurs périphériques multiplexés sur le même coupleur, l'organisation précédemment décrite ne peut plus s'appliquer telle quelle.

En effet, dans ce cas, il faut faire correspondre à un numéro de niveau et de sous-niveau plusieurs tables unités physiques associés aux différents périphériques multiplexés sur le même coupleur.

Pour résoudre ce problème on adoptera l'organisation suivante :

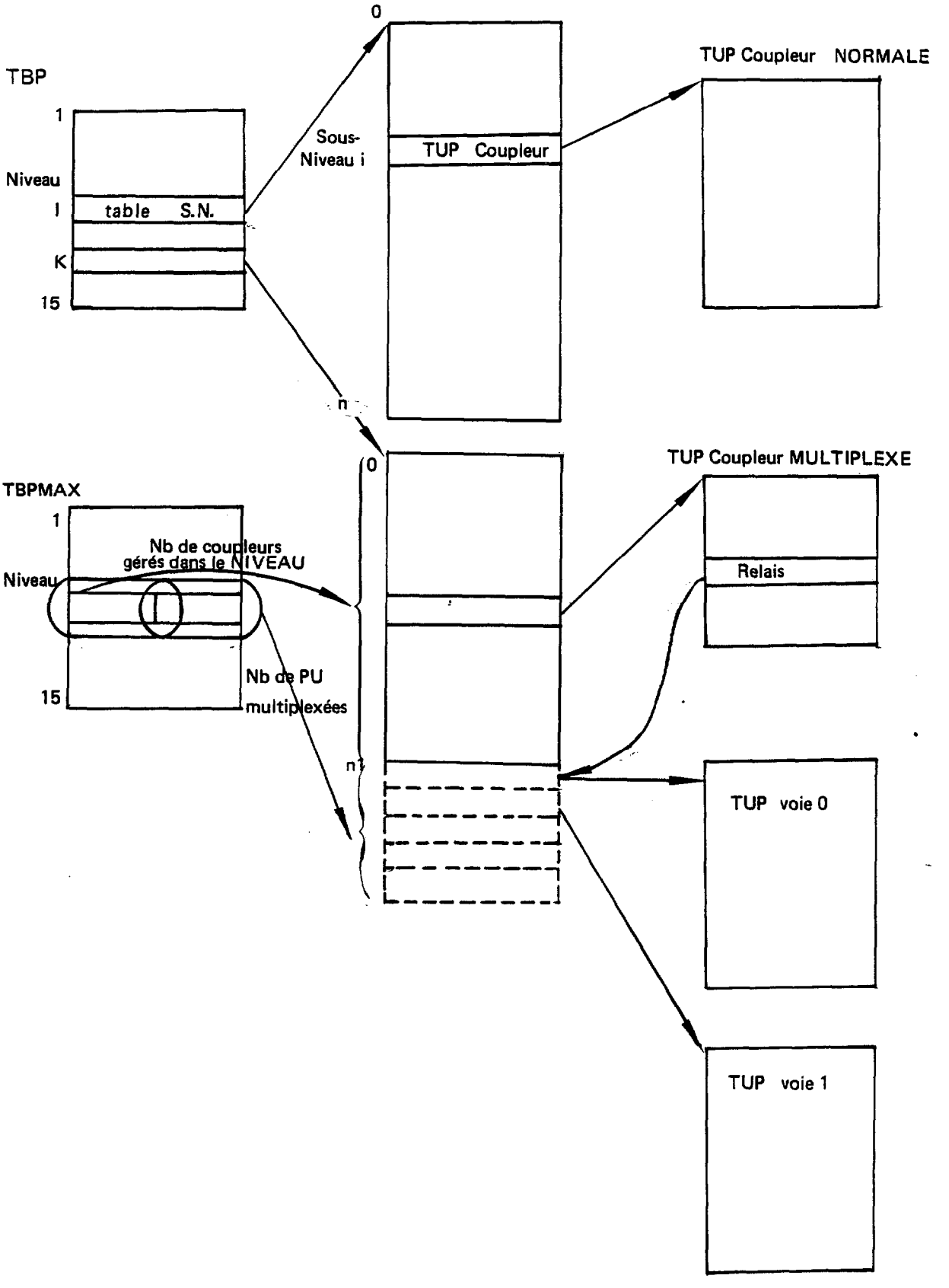
- **on associera à un coupleur multiplexé une table "unité physique fictive ou coupleur" dont l'emplacement est déterminé par la méthode ci-dessus. Cette table est gérée par IOCS sous-niveau hardware.**
- **on associera à chaque "voie" d'un coupleur multiplexé une table unité physique. Cette table est gérée par le driver en entretien d'échange.**

  Une table unité physique "coupleur" n'étant jamais directement associée à un échange, on peut lui donner une forme simplifiée.

Forme d'une table "unité physique fictive" :

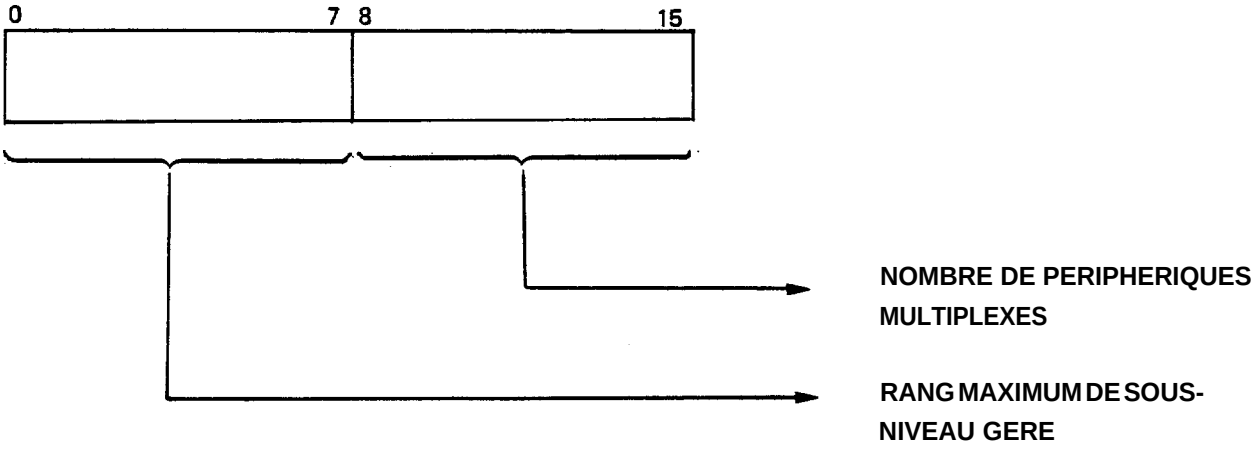
Mot (0)	STATUS
Mot (1)	MODEFO
Mot (2)	FUVOIE
Mot (3)	INDRIV
Mot (4)	ECHDRV
Mot (5)	BASEL
Mot (6)	VALSLO
Mot (7)	VALSLE
Mot (8)	RELAI
Mot (9)	FONC
Mot (10)	CONTEX
Mot)11)	ADMEM
Mot (12)	CONTOC
Mot (13)	ADPERI

Les mots (1), (2), (6). (7), (9), (10). (11), (12) sont sans signification pour une telle organisation.
Le mot (8) contient l'adresse d'une table indiquant les adresses des unités physiques pour chacune des voies.
Cette organisation permet de faire le lien entre une unité physique coupleur et les tables unités physiques associées aux différentes voies du coupleur multiplexé.


TBP
1
Niveau
I
table S.N.
K
15
0
Sous-
Niveau i
TUP Coupleur
TUP Coupleur NORMALE
n
TBPMAX
1
Niveau
Nb de coupleurs
gérés dans le NIVEAU
15
Nb de PU
multiplexées
0
n
TUP Coupleur MULTIPLEXE
Relais
TUP voie 0
TUP voie 1

1.3.7 - Table de description des niveaux hardware d'entrées-sorties (TBPMAX)

- cette table est organisée en mots
- elle est pointée par les priorités des tâches hardware
- elle contient pour chaque niveau géré :
  . le rang maximum du sous-niveau géré
  . le nombre de périphériques multiplexés dans le niveau.



1.3.8 - Table des fonctions spéciales moniteur - TBFSM

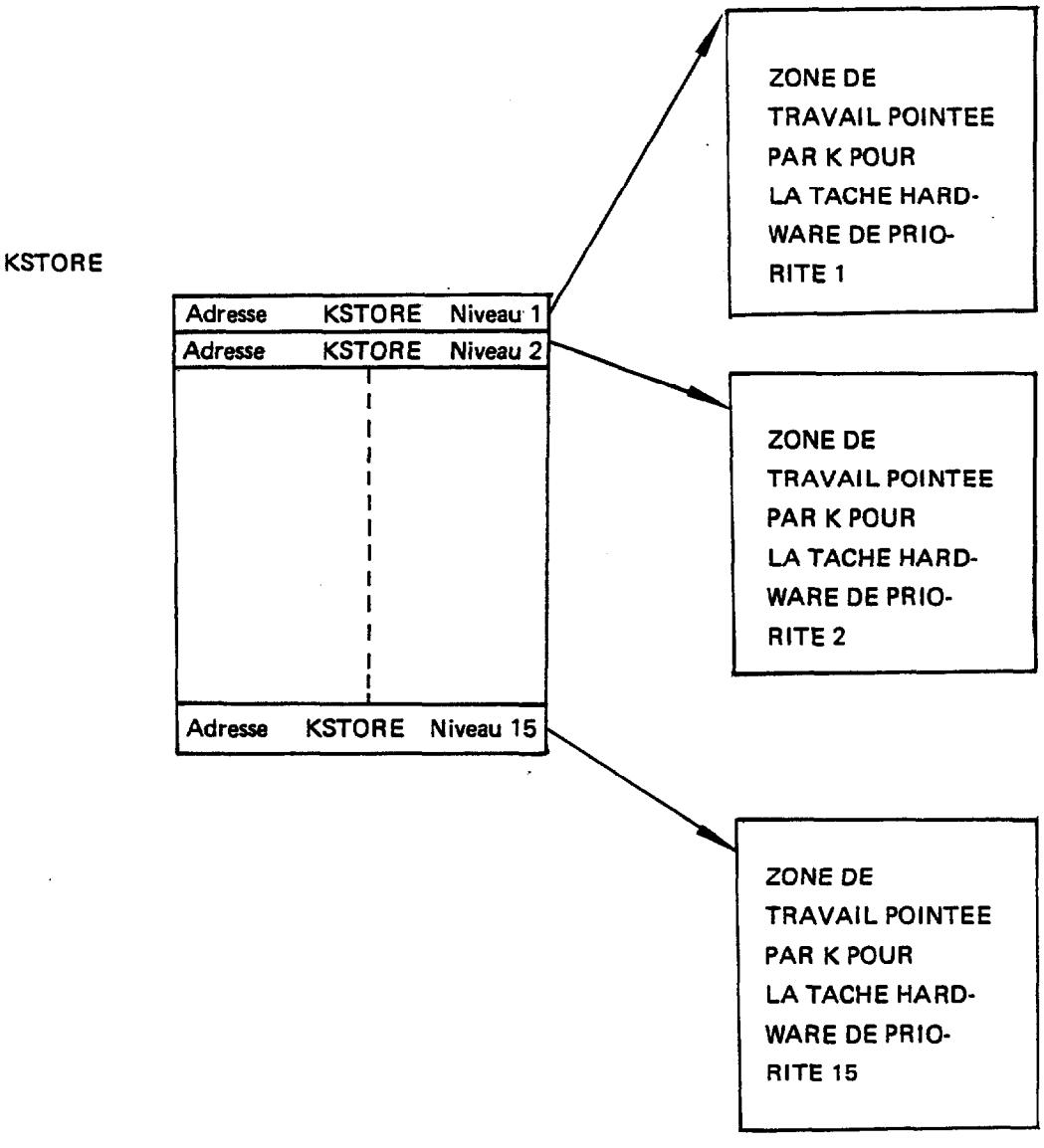
Cette table est organisée en mots. Elle contient les adresses des sous-programmes exécutant les fonctions spéciales adressées au moniteur.
Elle est pointée par les numéros des fonctions spéciales (bits 2 à 7 de l'octet de fonction).
En standard suivant la version d'IOCS celui-ci gère un certain nombre de fonctions spéciales.
L'utilisateur pourra intégrer à la configuration du système les fonctions qu'il jugera utiles.
Aux fonctions non existantes correspondront des adresses de programmes nulles.
La taille de la table est contenue dans la mémoire NFSMAX.

TBFSM

Adresse du module exécutant la fonction 0
Adresse du module exécutant la fonction 1
Adresse du module exécutant la fonction i

1.3.9 - Table des KSTORE des tâches hardware

Les tâches hardware gérées par IOCS ont leurs zones de travail pointées par le registre K dans les tables d'IOCS.



Nota : Pour un niveau non géré par IOCS, la zone de travail n'est pas réservée dans les tables et la position correspondante dans la table KSTORE vaut 0.

1.4 - LES TABLES DE GESTION DU DISQUE

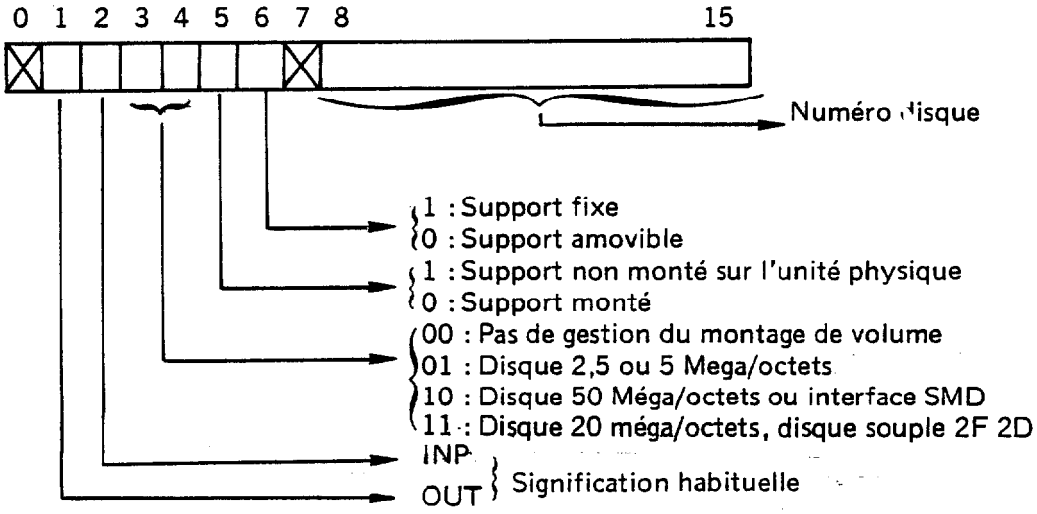
On rappelle qu'un disque peut être partagé en plusieurs unités fonctionnelles qui sont en fait des portions de disque (Voir Manuel de Référence).
Pour gérer une unité fonctionnelle disque, IOCS a besoin de connaître outre les paramètres qui caractérisent une unité fonctionnelle quelconque, les paramètres suivants :
- l'adresse disque du début de la zone affectée à l'unité fonctionnelle.
- le nombre de secteurs de cette zone.
Ces informations sont contenues dans des tables propres à la gestion du disque. Pour accéder à ces dernières on associe à chaque unité fonctionnelle disque un numéro de "disque" (les numéros allant de 0 à n - 1 si n est le nombre d'unités fonctionnelles disque).

1.4.1 - Numéro de disque

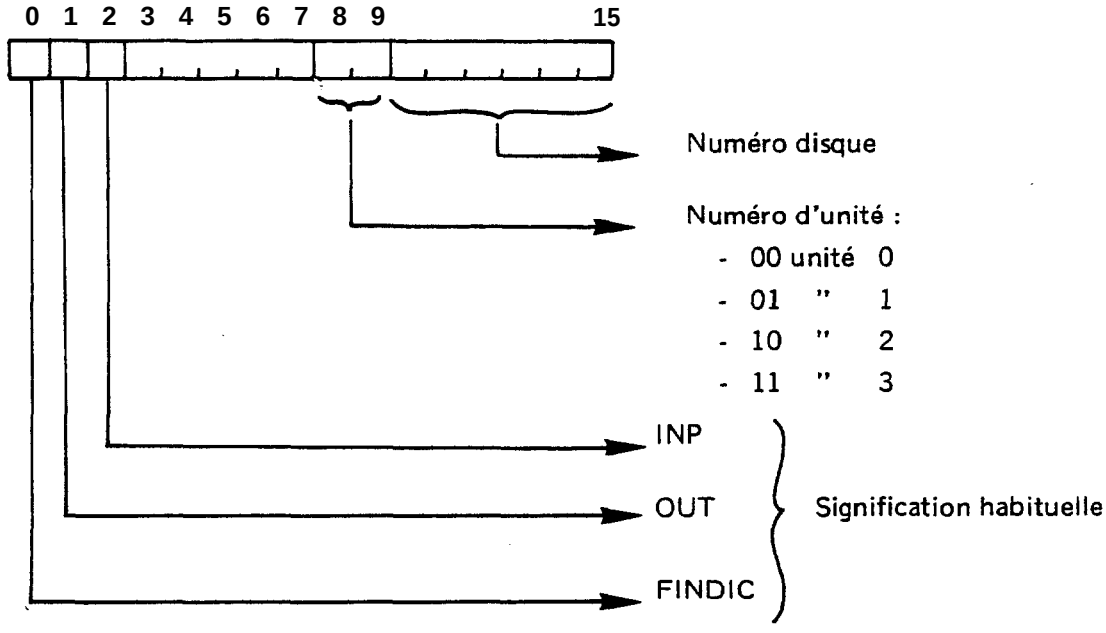
Pour les unités fonctionnelles autres que les unités fonctionnelles disque, on trouve dans la table TBMCOM un mot de commande qui est envoyé avant chaque échange par le driver.
Les échanges disque ne nécessitant pas l'envoi d'un mot de commande, on trouvera pour les unités fonctionnelles disque non pas un mot de commande mais un numéro de disque.

L'élément de la table TBMCOM correspondant à une unité fonctionnelle disque a le profil suivant :

a) pour les disques



b) pour les floppy-disques “multi-fu” 1 face, simple densité secteurs de 128 octets (driver utilisé : DRVIB)



1.4.2 - Table des adresses disque TBADK

- cette table est organisée en mots
- elle est pointée par les numéros de “disque”
- elle contient l’adresse sur le disque “réel” du début de la zone affectée à l’unité fonctionnelle disque
- sa taille est contenue dans la mémoire NDKMAX (nombre maximum d’unités fonctionnelles disque).

TBADK

Adresse du 1er Secteur du “disque” 0
Adresse du 1er Secteur du “disque” 1
Adresse du 1er Secteur du “disque” i

1.4.3 - Table des longueurs de disque TBLDK

- cette table est pointée par les numéros de “disque”
- elle contient le nombre de secteurs affecté à chaque unité physique
- sa taille est contenue dans la mémoire NDKMAX.

TBLDK

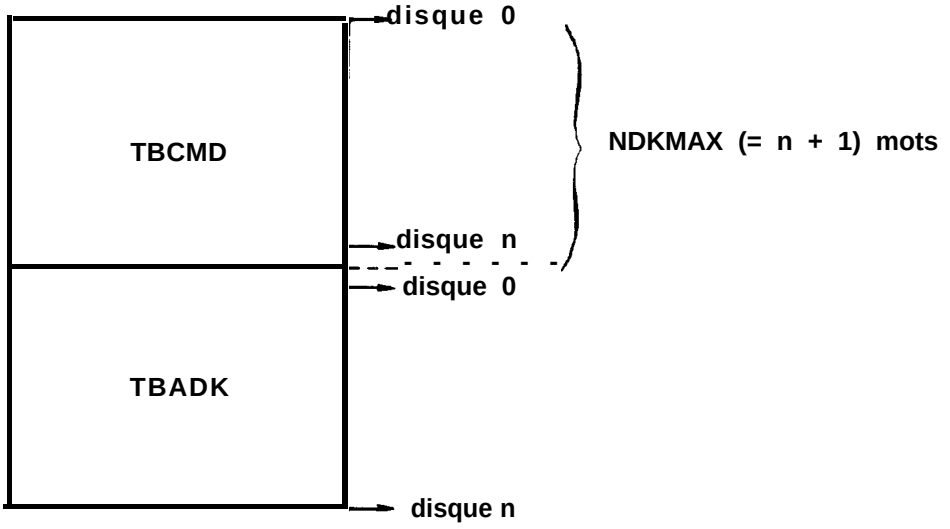
Nombre de secteurs du “disque” 0
Nombre de secteurs du “disque” 1
Nombre de secteurs du “disque” i

1.4.4 - Organisation des tables TBADK et TBLDK pour une organisation “grands disques”

Une organisation est dite grand disque quand l’adresse et la longueur disque sont sur deux mots. Cette organisation est utilisée dès qu’un disque de moyenne capacité a été décrit.
Les tables TBADK et TBLDK sont découpées en 2 sous-tables :
- la 1ère contenant les adresses ou les longueurs poids faible
- la 2ème contenant les adresses ou les longueurs poids fort.

1.4.5 - Table des mots de commande supplémentaire TBCMD

- cette table est organisée en mots
- elle est pointée par les numéros de disque
- elle est créée pour chaque FU “disque”
- elle contient un mot de commande supplémentaire
- elle se trouve devant la table TBADK et elle a la même longueur (NDKMAX).



- on accède à la table TBCMD par le contenu des mémoires TBADK et NDKMAX.
  pour le “disque” i

  TBCMD (i) = @ TBADK - NDKMAX + i

- dans le cas des disques souples gérés par le driver DRVFDD les bits 14 et 15 de TBCMD contiennent le numéro de voie (de 0 à 3).

2 - COMMENT ECRIRE UN DRIVER

Ce chapitre a pour but d'indiquer la procédure à suivre pour la réalisation d'un driver.

2.1 - DESCRIPTION GENERALE D'UN DRIVER

En règle générale on peut distinguer sept modules dans un driver :
- lecture mot d'état unité physique
- initialisation d'un échange effectif
- initialisation d'une fonction spéciale de positionnement
- entretien d'un échange effectif
- analyse des défauts
- initialisation de l'unité physique
- "tuer" un échange

Les modules "entretien d'un échange'* et "analyse des défauts" sont appelés par les tâches hardware qui gèrent les niveaux E/S. Tous les autres modules se déroulent sous le niveau software de l'appelant.

2.2 - LES MODULES D'UN DRIVER

Un module de driver a une structure de sous-programme. Généralement ces modules sont réentrants.
Tout driver doit disposer de deux points d'entrée.

2.2.1 - Entrée dite "INITIALISATION"

Ce point d'entrée est commun à plusieurs modules il est paramétré par le registre d'index.
- lecture mot d'état périphérique (PUSI) X = 0
- initialisation d'un échange (INIT) X = 1
- initialisation d'une fonction spéciale de positionnement (FSP) X = 2
- traitement des défauts (DEF) X = 3
- initialisation de l'unité physique (SCLEAR) X = 4
- tuer un échange (KILL) X = 5

2.2.2 - Entrée dite "ENTRETIEN"

C'est le point d'entrée dans un driver, sous niveau hardware, pour un périphérique fonctionnant en programmé prioritaire. Il est à noter que, pour un périphérique fonctionnant en mode canal, cette séquence ne sera jamais activée.

2.3 - FONCTIONS REALISEES PAR LES MODULES

2.3.1 - Lecture mot d'état

Le but-de cette fonction est de transmettre à l'appelant le mot d'état unité physique.

2.3.2 - Initialisation d'un échange

Ce module doit réaliser les fonctions suivantes :

- tester l'occupation du coupleur de façon à savoir s'il doit ou non poursuivre l'échange. Dans le cas où le périphérique est occupé à l'initialisation, le driver utilisera généralement la prochaine interruption pour relancer l'échange.
- en fonction des modalités de l'échange, il met à jour l'adresse du périphérique dans le mot ADPERI de la table d'unité physique
- pour les périphériques fonctionnant en programmé prioritaire, il met à jour dans le mot ECHDRV de la TUP l'adresse du module qui sera activé lors de la prochaine interruption pour l'entretien de l'échange.
- pour les périphériques fonctionnant en canal, il réserve un contexte canal (cette fonction est réalisée par le sous-programme d'IOCS SPCCN).
- dans le cas où le périphérique fonctionne en mode canal, le module initialise les registres du canal en vue de l'échange (cette fonction est réalisée par le sous-programme d'IOCS IPINI).
- il envoie le mot de commande associé à la FU qui pointe sur l'unité physique concernée (ce mot de commande peut être modifié par le module, en fonction des paramètres de l'échange).

2.3.3 - Entretien d'un échange

Ce module est activé pour les périphériques fonctionnant en programme prioritaire. Il doit assurer les fonctions suivantes :

- réaliser les opérations d'entrées-sorties proprement dites
- mettre à jour les données de l'échange dans la TUP (compte d'octets, adresse du prochain mot à transférer)
- déterminer le module à activer lors de la prochaine interruption et charger l'adresse de ce module dans le mot ECHDRV de la table d'unité physique.

2.3.4 - Initialisation d'une fonction spéciale de positionnement

Ce module se déroule sousniveau software, il doit interpréter et initialiser la fonction requise de la façon suivante :

- tester l'occupation du périphérique
- lancer la commande correspondant à la fonction demandée
- charger dans ECHDRV l'adresse du module qui sera lancé à la prochaine interruption.

2.3.5. - Traitement des défauts

1) Périphérique fonctionnant en programme prioritaire

Ce module devra lire le mot d'état périphérique et éventuellement le remettre en forme pour en faire un mot d'état "unité physique" et le transmettre à IOCS. Ce mot d'état doit être analysé de la façon la plus complète. En effet c'est à ce module de décider si l'échange doit ou non être interrompu. Si un défaut fatal est détecté le module positionnera BITDEF à 1 dans le STATUS de la TUP.

2) Périphérique fonctionnant en mode canal :

Après demande de compte-rendu canal (Sous-programme SPLIB d'IOCS), le driver détermine l'origine de l'interruption.

- fin d'échange : le compte d'octets restant à transmettre est nul. Il ne s'agit pas, alors, d'un défaut bien que délivré par une interruption exception.
- défaut : le compte d'octets restant à transmettre n'est pas nul. C'est alors au driver de décider si l'échange est, ou non, fatal.

En fin d'échange le contexte canal doit être libéré (Sous-programme LIBCCN d'IOCS).

2.3.6. - Initialisation de l'unité physique

Ce module doit initialiser les sémaphores de la TUP :

- sémaphore d'attachement (SEMATT) par la valeur 1
- semaphore d'exclusion (SEMGEN) par la valeur 1 et mise à 0 de la file d'attente.

Ces fonctions sont réalisées par le sous-programme INITP d'IOCS.

2.3.7. - "Tuer" un échange :

Cette fonction intervient lorsque l'on désire mettre fin à un échange. Suivant le type de périphérique, le driver prend l'initiative du traitement à effectuer.

Dans le cas où le périphérique dispose du mot de commande "RESET COUPLEUR", le driver l'envoie au coupleur. Dans le cas contraire, ou s'il n'est pas recommandé de l'envoyer (bande magnétique par exemple), le driver devra récupérer et acquitter les interruptions ultérieures intervenant hors échange.

Dans tous les cas avant le retour, le driver devra effectuer les traitements suivants :

- libération des ressources (retour à IOCS avec X=2)
- réveil de la tâche qui était en attente de fin d'échange.

Remarque : l'envoi de la commande "RESET COUPLEUR" inhibe les interruptions ultérieures sur celui-ci.

2.4. - INTERFACE IOCS - DRIVER

2.4.1. - Initialisation :

C = Valeur de la base C d'IOCS
L = Valeur de la base L du driver
W = adresse de la table d'unité physique
X = fonction à activer

PUSI	x = 0	(lecture mot d'état)
INIT	x = 1	(initialisation d'un échange effectif)
FSP	x = 2	(initialisation d'une fonction spéciale de positionnement)
DEF	X = 3	(traitement des défauts sous-niveau hardware)
CLEARS	X = 4	(initialisation d'une table d'unité physique)
KILL	X = 5	("Tuer" un échange).

Autres paramètres (IOCS - M2 seulement)

a) Appelant en mode maître et SVCS = 0
SLO = 0
SLE = 'FFFF (0 si pas de DRPS)
Y = adresse absolue de l'IOCB

b) Appelant en mode maître et SVCS = 1
SLO, SLE } appelant
Y = adresse absolue de l'IOCB

c) Appelant en mode esclave
SLO, SLE } appelant
Y = adresse absolue de l'IOCB si l'échange est en IBMOD (utilisation du pool buffer)
Y = adresse relative de IIOCB sinon.

Autres fonctions réalisées

Les informations suivantes, contenues dans la table d'unité physique, sont mises à jour par le noyau avant l'activation du driver :
- adresse de l'IOCB (ADIOCB) : même valeur que le registre Y
- adresse de la table d'échange (ADMEM) : cette adresse est absolue si :
  . l'appelant est en mode maître et SVCS=0
  . l'échange a lieu en IBMOD
  . le périphérique fonctionne en mode canal.

Elle est relative dans les autres cas
- adresse de la table des codes d'arrêt : il s'agit d'une adresse absolue si l'appelant est en mode maître et d'une adresse relative s'il est en mode esclave.
- numéro de FU (octet gauche de FUVOIE) sur lequel porte l'échange
- compte d'octets (CONTOC) : le bit 0 indique le sens de l'échange 0 : lecture, 1 : écriture
- valeur du registre SLO de l'appelant (VALSLO) : il a la valeur 0 pour un appelant en mode maître (avec SVCS = 0), il a la valeur du registre SLO de l'appelant sinon.
- valeur du registre SLE de l'appelant (VALSLE) : il a la valeur - 1 ('FFFF) pour un appelant en mode maître avec SVCS = 0, il a la valeur de SLE de l'appelant dans les autres cas.

Remarque : Dans le cas de IOCS - M1, toutes les adresses contenues dans la TUP sont des adresses absolues et VALSLO = VALSLE = 0.

Valeur de X	Signification du compte-rendu	Mesures prises par IOCS
0	Echange ou fonction de positionnement en cours	IOCS ne fait rien
1	Transmission d'un mot d'État sur les bits 5 à 15 de A avec éventuellement BITDEF = 1	- Mémorisation du mot d'état dans MODEFO - Si BITDEF = 1 abandon de l'échange (libération des ressources) avec compte-rendu éventuel dans le mot (3) de l'IOCB en cours
2	Défaut de fonctionnement du coupleur (exemple : coupleur répond occupe alors qu'il devrait être libre)	- Abandon de l'échange (libération des ressources) - Eventuellement compte-rendu dans le mot (3) de l'IOCB en cours
3	Défaut de fonctionnement du canal (exemple fin d'échange sur compte d'octets non nul)	- Abandon de l'échange (libération des ressources) - Eventuellement compte-rendu dans le mot (3) de l'IOCB en cours
4	Inutilisé	IOCS ne fait rien
5	Demande impossible à satisfaire dans la position actuelle du périphérique. Exemple : Demande d'écriture d'un bloc alors qu'on est en fin de bande sur un derouleur.	- Abandon de l'échange (libération des ressources) - Eventuellement compte-rendu dans le mot (3) de l'IOCB Le mot d'état transmis en compte-rendu sera le dernier mot d'état mémorisé dans MEDEFO
6	Fin d'échange normale ou fin de fonction de positionnement	- Libération des ressources - Eventuellement compte-rendu dans le mot (3) de l'IOCB
7	Paramètres incorrects	- Abandon de l'échange libération des ressources) - Eventuellement compte-rendu dans le mot (3) de l'IOCB

2.4.2 - Entretien

C = valeur de la base C d'IOCS
L = valeur de la base L du driver
W = adresse de la table d'unité physique
SLO - SLE = valeurs prises dans la table d'unité physique et décrites au paragraphe précédent.

Nota : Dans le cas de périphériques multiplexés la base W ne contient pas l'adresse de la table d'unité physique mais l'adresse de la table d'unité physique du coupleur (ou TUP coupleur). Il en est de même pour tes registres SLO et SLE qui doivent être initialisés avec les valeurs contenues dans VALSLO et VALSLE de la table d'unité physique de la voie. Tous les registres non sauvegardés par contexte, le sont systématiquement en début de la tâche hardware d'IOCS.

2.5 - INTERFACE DRIVER IOCS

- le module rend le contrôle à IOCS par une instruction RSR
- il peut restituer l'ensemble des registres dans un état quelconque exceptés :
  C : qui doit rester inchangé.
  W : qui doit contenir l'adresse de la table unité physique
  X : qui contient un compte-rendu dont la signification est donnée dans le tableau.
  En fin d'échange, le driver doit transmettre à IOCS, dans CONTOC, le compte d'octets restant à transmettre.

Nota : Le bit 0 doit toujours être à 0.

2.6 - INTEGRATION D'UN DRIVER A IOCS

Un driver doit être structuré comme un segment comportant :

- un local
- une section programme
- éventuellement une section table.

Il sera assemblé isolément puis relié à IOCS par édition de liens.
Il pourra utiliser le commun d'IOCS et devra alors déclarer cette section en DUMMY ou DSEC.

Règles à observer :

- le premier mot implanté est le premier mot du local du driver
- ce premier mot contient l'adresse de lancement initial du driver
- le deuxième mot du local contient la taille des TUP utilisés
  . octet gauche : taille de la TUP fictive (Mini : 14 mots)
  . octet droit : taille de la TUP vraie pour 16 tâches (mini 19 mots)
- les mots supplémentaires sont toujours au-dessus de la TUP

- les symboles à définir en ENTRY sont :
  - D R V x x x : pour le point d'entrée du driver initialisation
  - LOCx x x : pour l'adresse sur laquelle pointera la base L
  - E C H x x x : pour le point d'entrée du driver en traitement d'interruption (uniquement pour les périphériques multiplexés)
  - x x x : est le type de périphérique.

Exemple : Driver disque (OK)

```
                ENT  DRVDK
                ENT  LOCDK
                DSEC COM
COMMUN :        EQU  $ + 128
                |
                |
                |                  Description du COMMON d'IOCS
                |
                |
                DSEC TUP
STATUS :        W O R D  0
MODEFO :        W O R D  0
                |
                |
                |                  Description d'une table unité physique
                |
                |
                LOCAL
LOCDK :         EQU  $ + 128
                WORD DRVDK
                BYTE + i;+j
                |
                |
                |                  LOCAL du driver
                |
                |
                PROG
              USE C, COMMUN
              USE L, LOCDK
              USE W, STATUS
DRVDK :                            < POINT D'ENTREE DU DRIVER
                |
                |
                |
                |
                |
                |
                |
```

3 - COMMENT ECRIRE UN MODULE DE FONCTION SPECIALE MONITEUR

Un certain nombre de fonctions spéciales moniteur sont traitées en “standard” dans chacune des versions d’IOCS (voir manuel de référence IOCS).

Cependant, un utilisateur peut très simplement intégrer une nouvelle fonction spéciale à IOCS.

3.1 - INTERFACE IOCS - FONCTION MONITEUR ET FONCTION MONITEUR - IOCS

3.1.1 - Appel d’un module fonction spéciale

La table des adresses de programme de fonctions spéciales moniteur (TBFSM) permet à IOCS d’associer à chaque numéro de fonction spéciale une adresse de programme.

Cette table est chargée à la configuration d’IOCS. IOCS donne le contrôle à un module de fonction spéciale par une instruction BSR.

Il lui fournit à l’entrée les registres suivants :

C : pointe sur le COMMON
Y : pointe sur l’IOCB correspondant à la demande
W : pointe sur la table unité physique correspondant à la demande.

Remarque :
Dans le cas où un module de fonction spéciale a besoin d’une section LOCAL il devra lui-même charger sa base L contrairement à un module de driver. En effet IOCS ignore l’adresse du LOCAL associé à cette fonction spéciale.

3.1.2 - Retour à IOCS

- le retour à IOCS se fait par une instruction RSR
- l’état des registres peut être quelconque
- lorsqu’un module de fonction spéciale rend le contrôle à IOCS, celui-ci transmet à l’utilisateur le contenu du registre A tel qu’il l’a reçu. On peut donc utiliser le registre A pour transmettre à l’utilisateur une information ou un compte-rendu.

3.2 - INTEGRATION A IOCS

- un module de fonction spéciale peut être assemblé isolément puis relié à IOCS par édition de liens à condition que l'IOCS ait été configuré pour le recevoir
- le COMMON d'IOCS devra être décrit en tant que DSEC dans le module d'assemblage. La base C pointe cette DSEC (IOCS l'initialise avant de donner le contrôle à la fonction).
- le module pourra comporter une section LOCAL. Dans ce cas le programme devra lui-même charger sa base.

Règles à respecter :

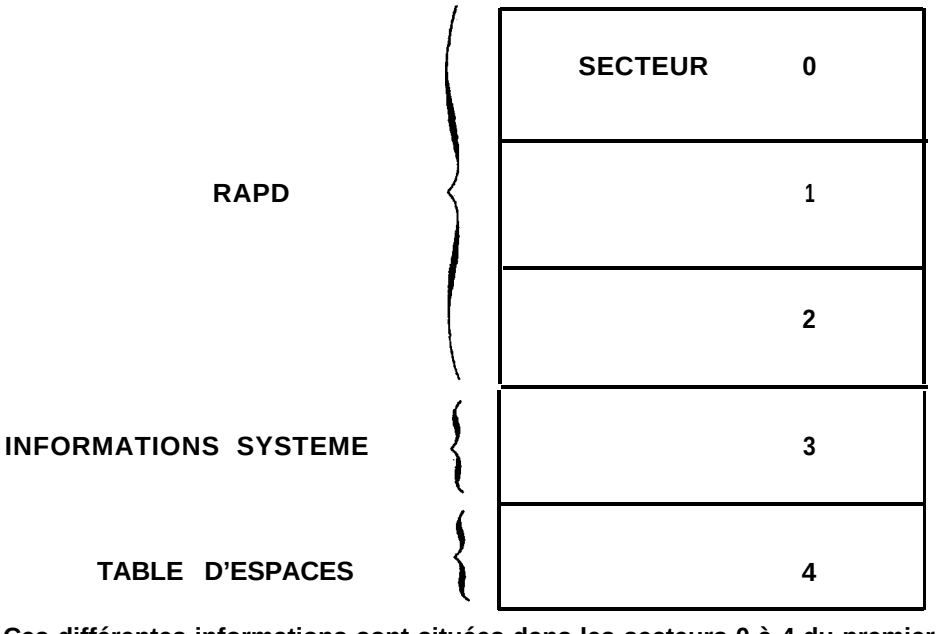
Le point d'entrée du module de fonction spéciale devra avoir pour nom symbolique F x x x x x où x x x x x représente le code mnémonique associé à la fonction lors de la configuration. Cette étiquette devra avoir été définie par une directive ENT.

4 - LE LOGICIEL DE GESTION DU MONTAGE DE VOLUME

4.1 - UTILISATION DES SUPPORTS DISQUE

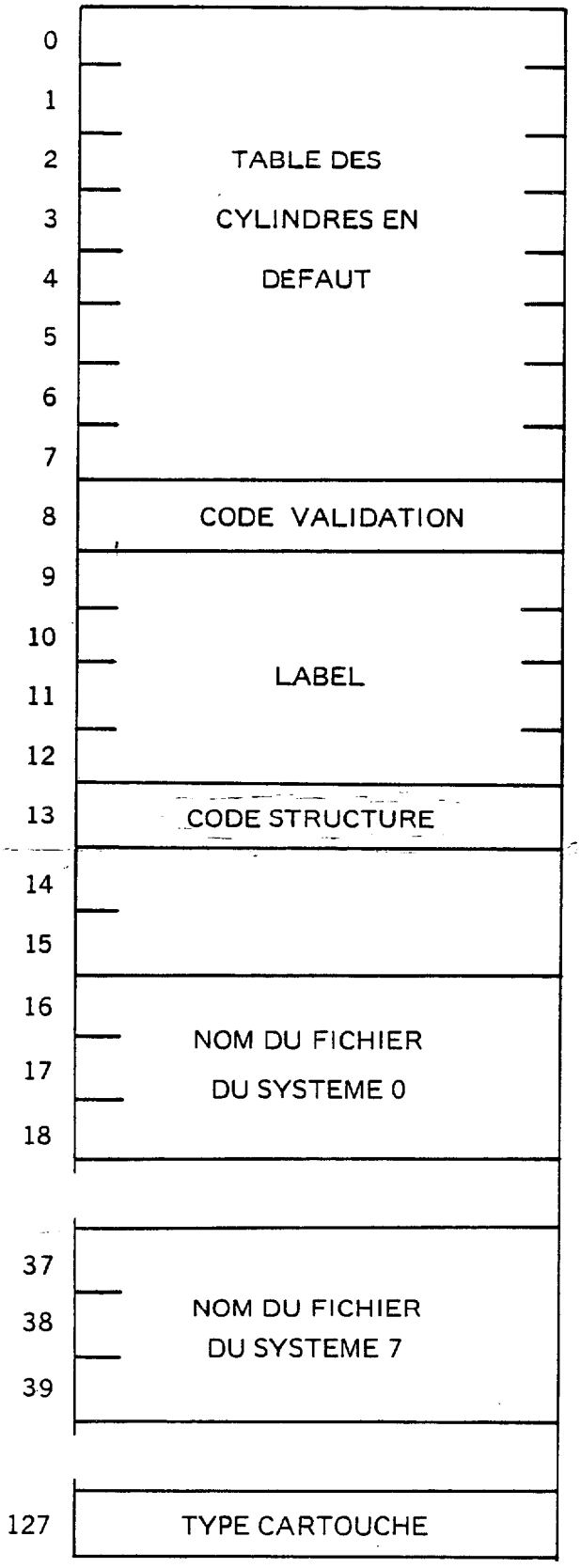
Avant toute utilisation un support disque doit être FORMATE. Cette opération, indispensable, réalisée par un programme autonome a pour but d'écrire sur le support des informations nécessaires au fonctionnement et à la gestion du support. Ces informations élaborées par le programme de formatage sont les suivantes :

- **les entêtes (HEADER) de secteurs nécessaires au fonctionnement du coupleur.**
- **la table des cylindres en défaut ou table de glissement indispensable au driver pour le calcul des adresses physiques.**
- **le programme de chargement et lancement (BOOTSTRAP) des systèmes disque (RAPD).**
- **un identificateur (label) du support.**
- **des tables systèmes renfermant différents indicateurs nécessaires à la prise en charge, du support par le système : TABLE D'ESPACES**



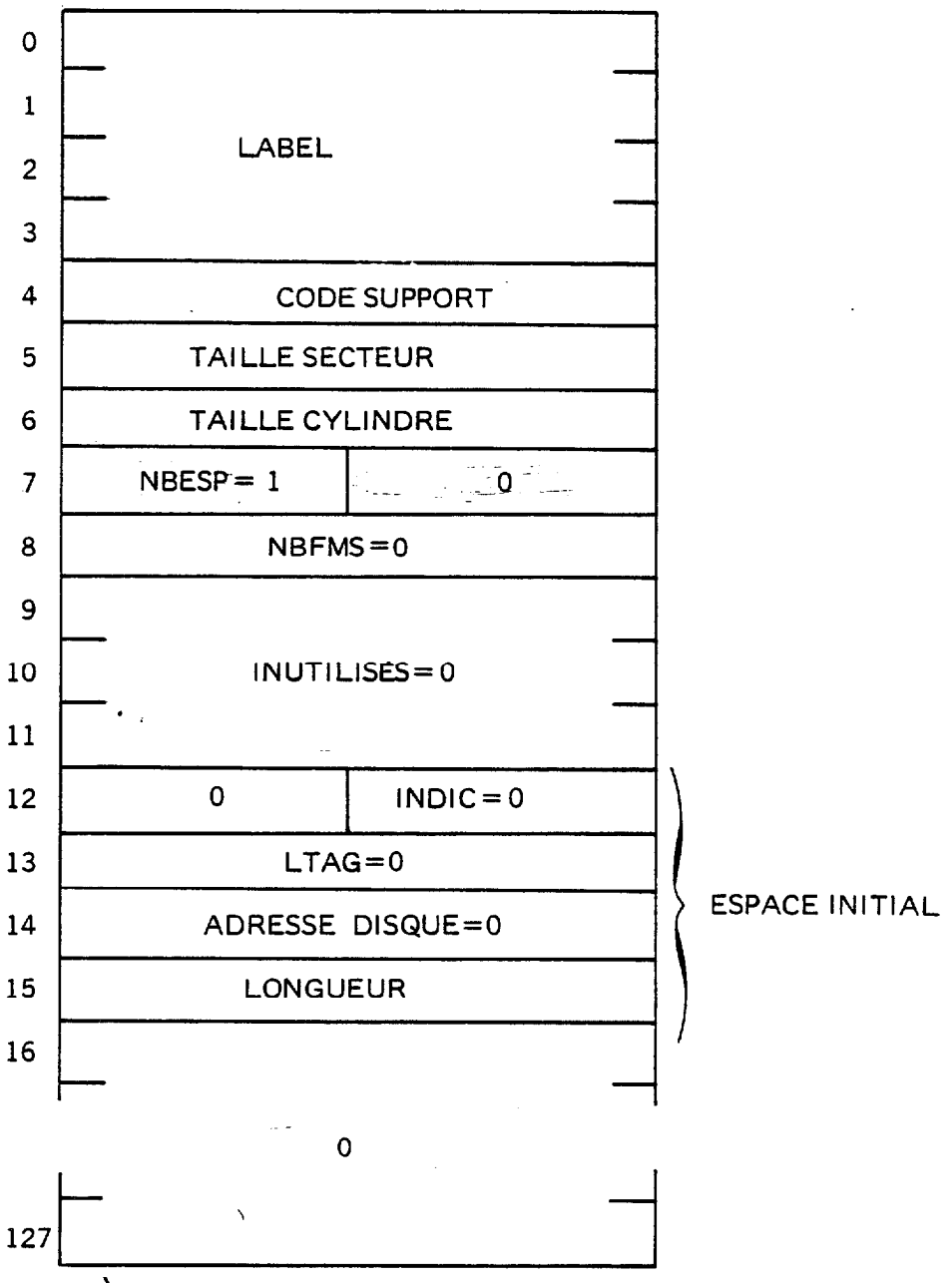
Ces différentes informations sont situées dans les secteurs 0 à 4 du premier cylindre valide du support.

4.1.1 - Structure du secteur 3



- **TABLE DES CYLINDRES EN DEFAUT : contient les numéros de cylindre en défaut ou '7FFF s'il n'y en a pas.**
- **CODE VALIDATION : '89AB**
- **LABEL : identificateur du support (1 à 7 caractères ASCII, suivi du caractère '8D).**
- **CODE STRUCTURE :**
  - *** 0 : volume non structuré (secteur 4 sans signification).**
  - *** 1 : volume structuré et le support est un disque à cartouche.**
  - *** 2 : volume structuré et le support est un disk pack.**
  - *** 4 : volume structuré fixe**
  - *** 5 : volume structuré mobile** **le support est un disque 20 mb**
  - *** 6 : volume structuré et le support est un floppy disk 2F 2D**
  - *** 7 : volume structuré, le support est un disque à interface SMD.**
  - *** 8 : volume structuré, le support est un disque WINCHESTER.**
- **La zone contenant les noms des fichiers systèmes n'est renseignée que pour les supports système.**
- **TYPE CARTOUCHE :**
  - *** Q : cartouche banale.**
  - *** n : cartouche de régénération. n = N° du système REGEN.**

4.1.2 - La table d'espaces après formatage (secteur 4)

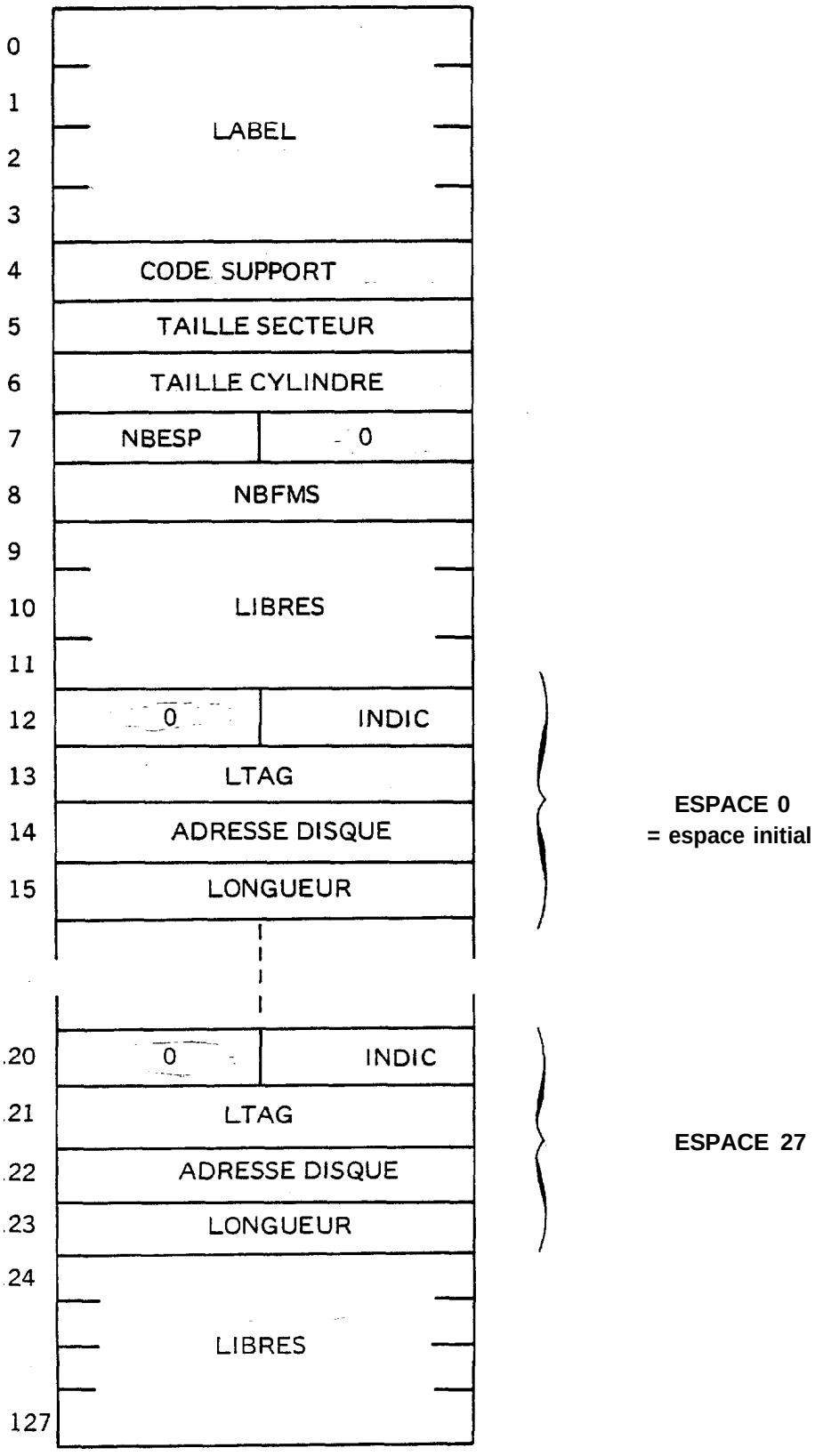


LABEL:
CODE SUPPORT : } **recopie des informations contenues dans le secteur 3.**

TAILLE SECTEUR : Taille du secteur en mots (128).
TAILLE CYLINDRE : Taille du cylindre en secteur (fonction du support).
NBESP: 1
NBFMS : 0
INDIC : 0 (Espace accessible en lecture et écriture)
LTAG : 0 (Espace non géré par FMS)
ADRESSE DISQUE : 0
LONGUEUR : nombre de cylindres formatés.

4.1.3 - La table d'espaces structurée par FUP4 (SDEF).

Après avoir formaté le support, l'utilisateur dispose d'un disque susceptible d'être "monté" sur une unité physique.
Préalablement à tous accès il doit compléter la structure définie par le programme de formatage au moyen de l'utilitaire FUP4 (voir notice FUP) et définir l'environnement du système de fichiers FMS à l'aide du même utilitaire. Le secteur 4, lorsque cette opération a été réalisée, se présente ainsi :



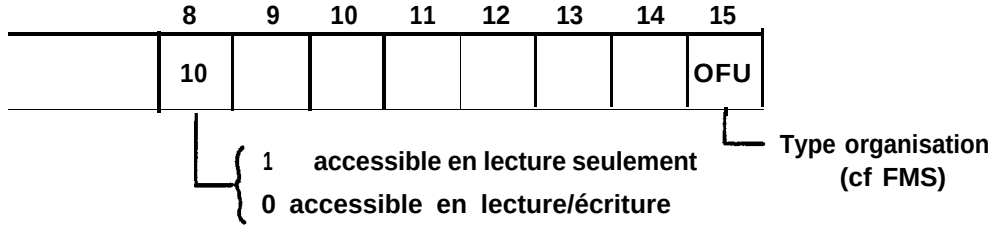
Signification des différentes valeurs :

a) Paramètres communs à l'ensemble des espaces

LABEL :
CODE SUPPORT : } **recopie des informations contenues dans le secteur 3.**
TAILLE SECTEUR : Taille du secteur en mot (128).
TAILLE CYLINDRE : Taille du cylindre en secteur (fonction du support).
NBESP : Nombre d'espaces valides décrits sur le support.
NBFMS : Nombre d'espaces disque gérés par FMS (cf notice FMS).

b) Paramètres relatifs à un espace

INDIC :



LTAG : Longueur de la Taille d'Allocation des Granules.
si LTAG = 0 l'espace n'est pas géré par FMS.
ADRESSE DISQUE : Adresse physique, exprimée en cylindres, de l'espace.
LONGUEUR : Taille de l'espace disque exprimée en cylindre.

4.2 - UTILISATION DU LOGICIEL

4.2.1 - Description interne

Le logiciel de gestion du montage des volumes se présente sous la forme de trois fonctions spéciales moniteur (cf manuel de référence) :

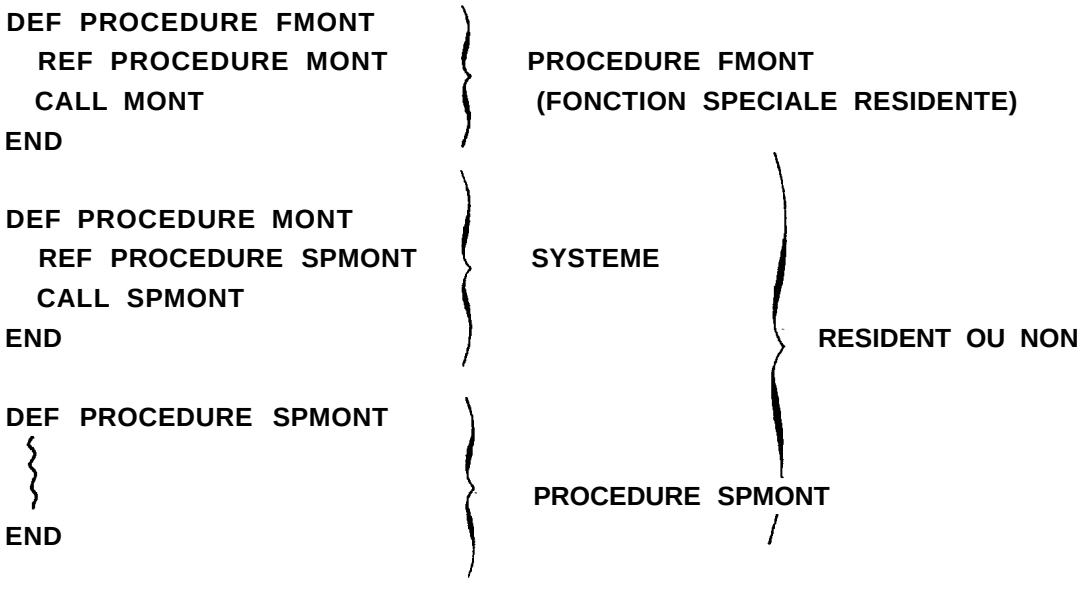
- FREAD : cette fonction spéciale a pour but de communiquer au système et aux processeurs la structure du support monté sur une unité physique. Cette requête appelée par les systèmes et par certains processeurs est RESIDENTE.

- FMONT et FDMONT : ces fonctions spéciales accessibles à travers les dialogues opérateur du système sont activées uniquement lors des “montages” et “démontages” des volumes. Il est donc possible de les rendre NON RESIDENTES et de les induredans la structure d’OVERLAY du système. Pour cette raison chaque fonction spéciale FMONT et FDMONT se compose de 2 modules :

* un module résident constituant la fonction spéciale moniteur dont le seul rôle est de réaliser un appel vers un sous-programme système qui appellera lui-même le sous-programme effectuant le traitement.

* le module ou sous-programme de traitement proprement dit.

Exemple : Fonction spéciale FMONT



4.2.2 - La bibliothèque BIBVOL

COMPOSITION :

La bibliothèque BIBVOL support du logiciel comporte 5 articles. C’est un fichier indexé catalogué sous le nom de BIBVOL - : S ; les articles constituant ce fichier sont :

- FREAD : fonction spéciale moniteur de communication de structure disque.
- FMONT : fonction spéciale moniteur de demande de montage de volumes.
- SPMONT : module complément du précédent.
- FDMONT : fonction spéciale moniteur de demande de démontage de volumes.
- SPDMON : module complément du précédent.

UTILISATION

La scrutation de la bibliothèque à l'édition de liens du système doit intervenir :
- à l'édition de liens de la racine pour les fonctions spéciales moniteur FREAD, FMONT, FDMONT.
- à l'édition de liens de la racine pour les sous-programmes SPMONT et SPDMON dans le cas où ces fonctions doivent être résidentes.
- à l'édition de liens des branches gérant le montage et démontage de volumes dans le cas où ces fonctions doivent être à structure d'overlay.

4.3 - GENERATION

L'utilisateur demande l'intégration de l'option gestion de volume à la génération du système en utilisant les macros de GENIO prévues à cet effet.

4.3.1 - Description des macros

Elles permettent de décrire :

a) Les fonctions spéciales moniteur non standard

Les trois fonctions spéciales moniteur non standard doivent être link-éditées avec IOCS, lorsque l'option GESTION DE VOLUME est choisie.
Ces fonctions, non standard font partie de la bibliothèque BIBVOL. Ce sont :
- **FDMONT**
- **FMONT**
- **FREAD.**

Elles peuvent être déclarées séparement par l'intermédiaire des macros :

```
%FSMON   DMONT   NUM   =   8
%FSMON   MONT    NUM   =   9
%FSMON   READ    NUM   =   10
```

Cependant l'utilisateur peut utiliser la macro :

```
%FSMON     GVOL
```

qui générera automatiquement les 3 macros précédentes.

b) Les unités d'échange multiplexé "disque"

Elles sont décrites par les macros %PUCD, %PUVM, %PUDP et CPSMD (cf chapitre Description de la Bibliothèque des macro-instructions).
La description des paramètres des macros sont les mêmes pour les systèmes utilisant ou n'utilisant pas l'option gestion de volume.

c) Les unités fonctionnelles disques

Les macros de génération disponibles sont conçues afin de permettre à l'utilisateur de générer des FU disque statiques, des FU disque dynamiques standard et des FU initiales.

Générer une FU statique consiste à fixer à la génération la valeur du couple ADRCYL/NBCYL qui définit la FU. Les macros permettant de générer des FU statiques sont %FUCD et %FUDP (cf chapitre description de la bibliothèque des macro-instructions).

Générer une FU dynamique standard consiste à ne donner aucune valeur au couple ADRCYL/NBCYL. Ces informations seront chargées par le système lors de la reconfiguration dynamique des tables d'IOCS avec les valeurs inscrites sur le support dans la table d'espace.
Les macros prévues pour générer facilement des FU dynamiques standard sont FUESPCD et FUESPDP (cf chapitre description de la bibliothèque des macro-instructions). Elles génèrent des FU fermées (ADRCYL = 0 et NBCYL = 0) sur lesquelles aucun échange n'est possible avant la reconfiguration dynamique (MONT sous BOS/D).

Générer une FU initiale consiste à décrire la première FU de l'unité physique. Une FU initiale est générée ouverte (ADRCYL = 0 et NBCYL ¹ 0) de manière à accéder au cylindre 0 du support disque dans les fonctions spéciales de montage de volume.
Les macros FUICD et FUIDP permettent de générer des FU initiales avec ADRCYL = 0 et NBCYL = 400 (cf chapitre description de la bibliothèque des macro-instructions).

Remarques :

- Il est toujours possible de décrire un disque avec gestion d'espace par l'intermédiaire des macros de génération des FU statiques.

- Pour décrire un disque avec gestion d'espace on peut utiliser simultanément les macros de génération des FU statiques, des FU dynamiques standard et de la FU initiale.

- Pour générer une FU initiale pour un disque 200 cylindres il faut utiliser les macros de génération des FU statiques.

4.3.2 - Exemple de génération

Les exemples ci-dessous reprennent les exemples décrits dans le manuel de référence IOCS.

Exemple 1 :

Génération d'une unité physique sur laquelle il y aura reconfiguration dynamique. On veut générer les unités fonctionnelles D9 (FU initiale) D3, D5, D6, E1, E2.
La solution préconisée est la génération dynamique standard, par l'intermédiaire des macros de génération dynamique.

```
%FUICD  D9  VOIE  =  0  FIXE  =  N
%FUESPCD   D3
%FUESPCD   D5
%FUESPCD   D6
%FUESPCD   E1
%FUESPCD   E2
```

Exemple 2 :

Génération d'un disque avec gestion d'espace mais sans reconfiguration (ex : le disque système sous BOS/D) ; elle se fait par l'intermédiaire des macros de génération statique.

```
%FUCD  D1  VOIE  =  0  ADRCYL  =  000  NBCYL  =  40  FIXE  =  Y
%FUCD  D2  VOIE  =  0  ADRCYL  =  040  NBCYL  =  242  FIXE  =  Y
%FUCD  D4  VOIE  =  0  ADRCYL  =  282  NBCYL  =  8  FIXE  =  Y
```

Exemple 3 :

Génération d'un disque avec gestion d'espace et sans reconfiguration mais pour lequel on veut avoir des macros de génération compatibles entre différents systèmes (ex : le disque système pour BOS et TSM). Ce mode de génération est nommé gestion d'espace dynamique préinitialisée. Elle utilise les macros de génération statique (pour définir les FU système D1 - D2 et celle utilisée sous BOS/ D4) et les macros de génération dynamique (pour les FU utilisées sous TSM : D8).

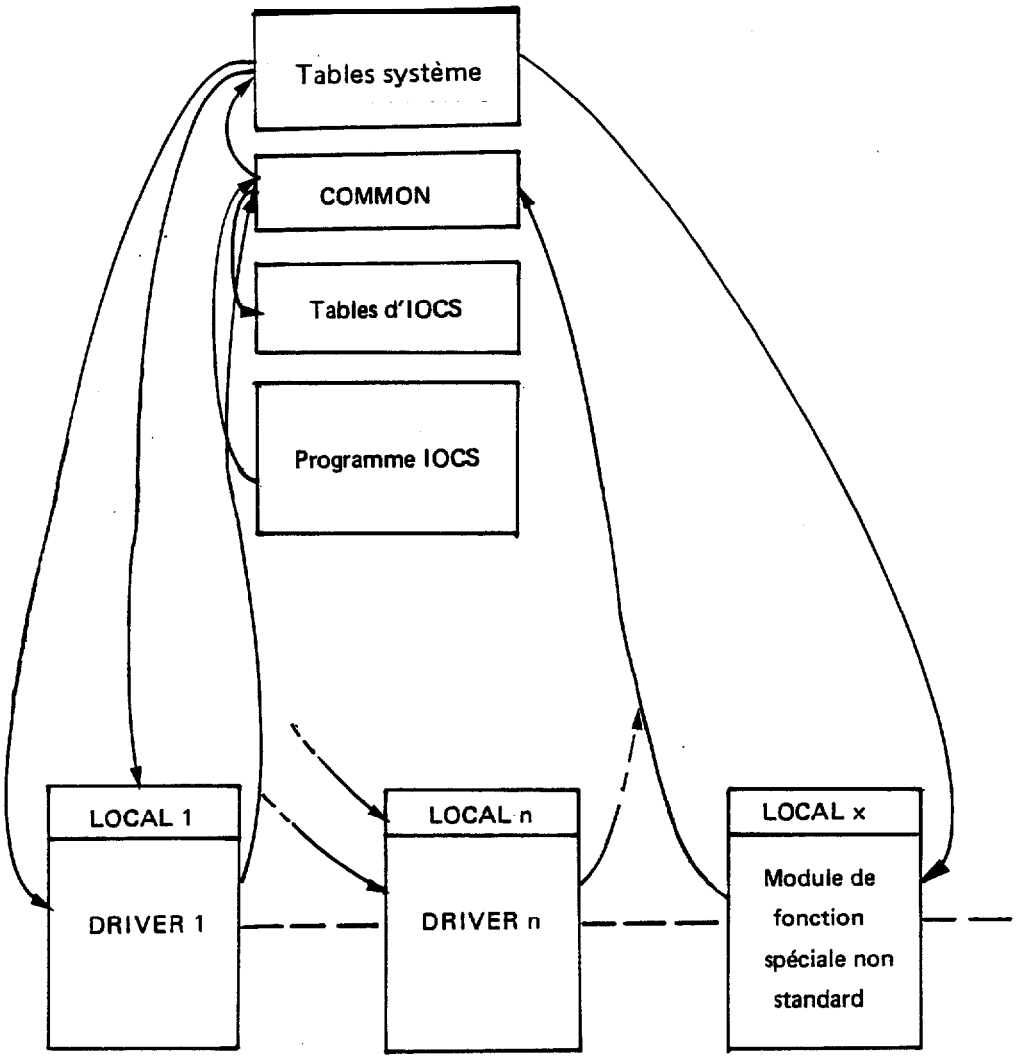
```
%FUICD  D7  VOIE  =  0  FIXE  =  Y
%FUCD  D1  VOIE  =  0  ADRCYL  =  0  NBCYL  =  40  FIXE  =  Y
%FUCD  D2  VOIE  =  0  ADRCYL  =  40  NBCYL  =  242  FIXE  =  Y
%FUCD  D4  VOIE  =  0  ADRCYL  =  282  NBCYL  =  110  FIXE  =  Y
%FUESPCD   D8
```

DEUXIEME PARTIE :

CONFIGURATION D'IOCS

IOCS est un programme modulaire. Chaque utilisateur à la possibilité de configurer un IOCS approprié à son installation (système d'entrées-sorties) et à ses besoins. Pour cela il dispose de GENIO (générateur IOCS).

1 - INTRODUCTION STRUCTURE D'IOCS



1.1 - LE NOYAU D'IOCS

Certains éléments sont indépendants de la configuration du système et sont nécessaires au fonctionnement d'un IOCS destiné à une configuration minimale. Ils constituent le "noyau" d'IOCS.

Le noyau d' IOCS comprend :

- le COMMON
- les tables d'IOCS
- le programme IOCS

Le noyau d'IOCS est indépendant de la configuration du système et n'a donc pas besoin d'être "généré". Il est fourni sous une forme symbolique directement assemblable par ASM.

On sait (voir Manuel de référence) qu'il existe trois versions d'IOCS :

- une version "mono-tâche" utilisable sur le SOLAR 16 sans ou avec l'option scheduler microprogrammé
- 2 versions "multi-tâches" utilisables uniquement sur le SOLAR 16 avec l'option scheduler microprogrammé.

Il existe donc trois versions du noyau d'IOCS
- un noyau IOCS mono-tâche : IOCS-S
- deux noyaux IOCS multi-tâches. Ces deux versions ont les mêmes caractéristiques. Elles se différencient par la taille mémoire gérée : - 32 K maximum pour la version IOCS-M1
  - plus de 32 K pour la version IOCS-M2.

1.2 - ELEMENTS DEPENDANT DE LA CONFIGURATION DU SYSTEME

Ce sont :

- les tables système
- les drivers
- les modules de fonctions spéciales non standard.

1.2.1 - Les tables système

Elles sont décrites dans la première partie du présent manuel.

Elles renferment toutes les informations caractérisant d'une part le système d'entrées-sorties (unités physiques, unités fonctionnelles, unités symboliques), d'autre part les fonctions d'IOCS.

Ces tables sont évidemment particulières à une application donnée et nécessitent donc d'être générées. C'est le but de GENIO.

1.2.2 - Les drivers

Chaque driver est structuré comme un segment utilisant d'une part le COMMON d'IOCS, d'autre part un local qui lui est propre.

Chaque driver constitue un module assemblé isolément et link-éditable avec IOCS.

Les drivers des périphériques conventionnels sont fournis sous une forme binaire link-éditable. On reliera à IOCS uniquement les drivers nécessaires à l'installation.

Dans le cas où l'utilisateur veut introduire un périphérique de type nouveau, il pourra écrire un driver approprié à ce périphérique. Ce driver pourra également être relié à IOCS par édition de liens comme cela est fait pour les périphériques conventionnels, à condition que ce nouveau type de périphérique ait été défini lors de la configuration.

1.2.3 - Les modules de fonctions spéciales

Certaines fonctions spéciales dites standard sont incluses dans le noyau d'IOCS. Si l'utilisateur veut en introduire d'autres, il devra écrire le module correspondant (voir paragraphe 3 première partie).

Ce module pourra être relié à IOCS par édition de liens comme cela est fait pour les drivers.

2 - LES DIFFERENTES PHASES DE LA CONFIGURATION

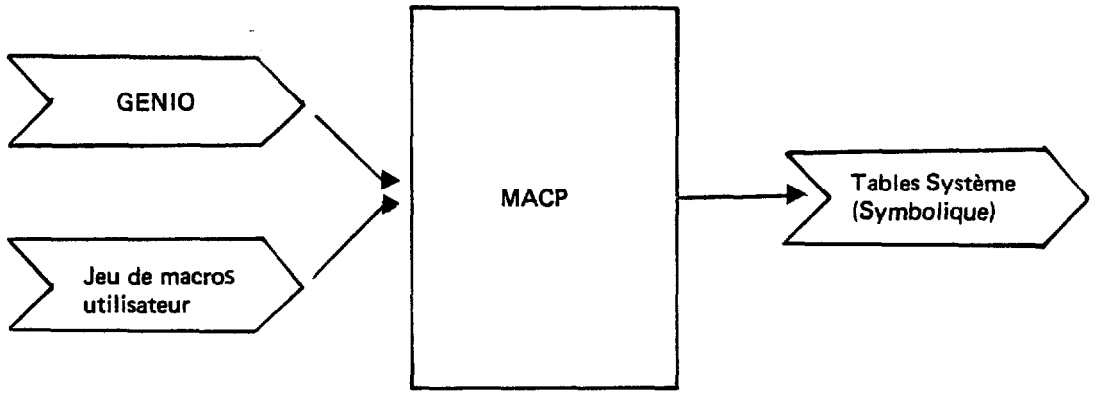
Pour obtenir une bande binaire translatable de l'ensemble IOCS + drivers + fonctions non standard, trois phases sont en général nécessaires.

2.1 - GENERATION DES TABLES SYSTEME

Cette première phase utilise le macroprocesseur MACP, elle est réalisée par GENIO :

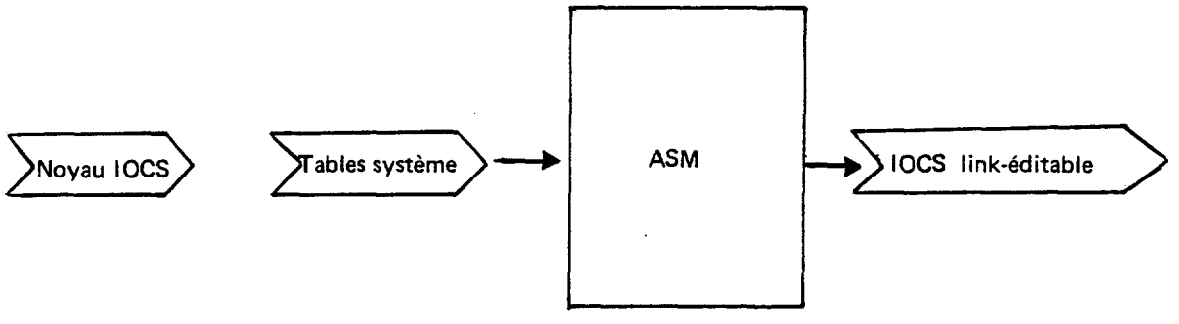
- L'utilisateur peut décrire son système dans un langage extrêmement simple en écrivant un jeu de macro instructions.

A l'issue de cette première phase, l'utilisateur dispose d'une bande en langage symbolique assemblable par ASM décrivant les paramètres du système et les tables système.



2.2 - ASSEMBLAGE

Dans cette deuxième phase l'utilisateur devra assembler successivement la bande "Tables système" puis le "noyau d'IOCS".

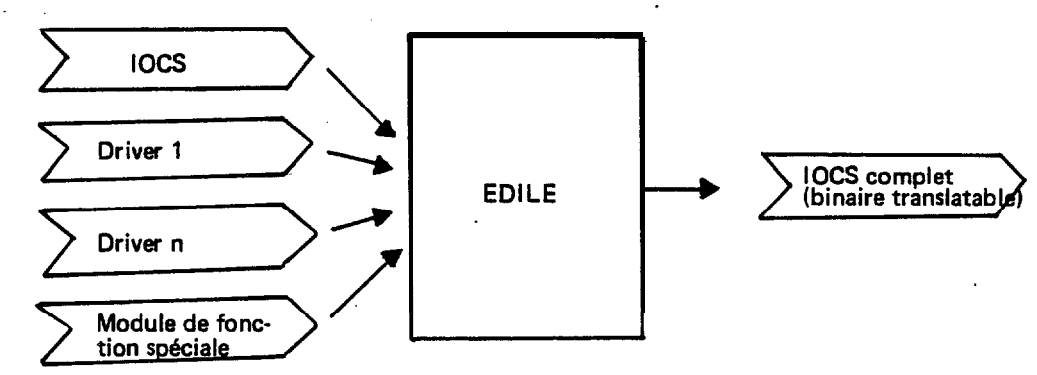


A l'issue de cette deuxième phase, l'utilisateur dispose d'une bande binaire d'IOCS qu'il peut éventuellement link-éditer avec les drivers des périphériques...

Remarque :

Lors de l'assemblage, on devra obligatoirement assembler la bande "Tables système" avant le "Noyau d'IOCS". Cette bande comporte en effet un certain nombre de définitions de symboles valeur utilisés par le noyau d'IOCS.

2.3 - EDITION DE LIENS



Remarque :

Le driver horloge temps réel (DRVHTR) est parfois amené par le superviseur (BOS-D, RTES-D). L'IOCS de ces systèmes ne pourra être obtenu indépendamment sous forme de binaire translatable. Dans ce cas l'utilisateur devra link-éditer globalement IOCS, les drivers, éventuellement les modules de fonction spéciale et le superviseur.

3 - GENERATION

3.1 - PRESENTATION DE GENIO

GENIO se présente sous la forme d'une bande ou fichier symbolique comportant :

- une séquence d'initialisation
- une bibliothèque de macro-définitions

Pour décrire sa configuration, l'utilisateur devra écrire un jeu de macro-instructions.
Le contenu de la bibliothèque sera décrit ultérieurement.

3.2 - CARACTERISTIQUES

La génération est conversationnelle.
La plupart des erreurs d'écriture des macroinstructions (erreurs de syntaxe, paramètres incorrects, . . . etc) peuvent être corrigées immédiatement car elles sont détectées lors de la génération.

3.2.1 - Les messages d'erreurs

Les messages d'erreurs peuvent être de deux sortes :

a) Messages émis par MACP

Ces messages sont de la forme ERM n
< Macroinstruction erronée >

Pour connaître les différentes causes d'erreurs, se reporter au manuel de référence de MACP.

On rappelle que les erreurs dans l'écriture d'une macroinstruction ne peuvent être détectées que si elle est précédée par le caractère %. Pour des raisons de sécurité, il est donc conseillé de faire précéder toute macroinstruction par le caractère % (en colonne 1).

b) Messages émis par GENIO

Lorsque GENIO trouve une erreur dans l'écriture d'une macroinstruction, il tue la macro-instruction erronée et imprime un message d'erreur.

Les différents messages sont :

- MACROINSTRUCTION INTERDITE :
  La macroinstruction écrite précédemment n'est pas autorisée dans l'étape actuelle : elle ne respecte pas l'ordre demandé
- PARAMETRE ? INCORRECT :
  Le paramètre spécifié par ? n'est pas correct. Pour plus de précisions se reporter à la description de la macroinstruction en cause
- INCOMPATIBILITE ENTRE ? ET ? :
  Les deux paramètres ne sont pas compatibles.
- PARAMETRES ? , ? IDENTIQUES
  Les 2 paramètres doivent être différents.

Des détails concernant les différentes causes d'erreurs sont donnés dans la description de chaque macroinstruction.

3.2.2 - La correction des erreurs

Lorsqu'une erreur a été détectée en cours de génération, soit par MACP, soit par GENIO, MACP lit la suite du programme source non plus sur l'unité symbolique "SI" mais sur l'unité symbolique "EC".

En général, on affectera l'unité symbolique "EC" au clavier du télétype, ce qui permet de corriger immédiatement la macroinstruction erronée en frappant la phrase correcte sur le clavier.
L'utilisateur peut alors introduire autant de phrases qu'il le désire.
Pour retourner à l'unité symbolique "SI" on frappera une ligne vide (CR) .

4 - DESCRIPTION DE LA BIBLIOTHEQUE DE MACROINSTRUCTIONS

4.1 - NOTATIONS

XXX	Chaîne de caractères de type "symbole" dont le nombre de caractères est limité au nombre de X.
999	Chaîne de caractères de type "décimal" dont le nombre de caractères est limité au nombre de 9.
FFF	Chaîne de caractères de type "hexadécimal" dont le nombre de caractères est limité au nombre de F.
{---- {---- {----	Indique un choix entre plusieurs possibilités
[----	Indique un paramètre qui peut être omis.

4.2. - STRUCTURE D'UN JEU DE MACROINSTRUCTIONS

Lors de l'écriture d'un jeu de macroinstructions, l'utilisateur doit respecter un certain ordre. Toute macroinstruction qui ne respectera pas cet ordre sera tuée par MACP et le message "MACROINSTRUCTION INTERDITE" sera imprimé.

On distingue trois grandes phases :

4.2.1 - Initialisation

Cette phase permet de préciser la version d'IOCS avec laquelle on désire travailler. Elle est constituée par la macroinstruction %ØPTION = ?
Cette macroinstruction est obligatoire à la tête de tout jeu d'instruction.

4.2.2 - Dimensionnement et définitions

Cette phase permet de fixer les paramètres du système et de définir les périphériques non standard présents dans la configuration. Les macro-instructions autorisées sont :

%NTACHES
%SUMAX
%INPMAX
%POOL
%FSMON
%DEFPU
%SYMBEXT

Pour décrire les systèmes “standard” on peut utiliser les macro-instructions :

%BOSD
%RTESM
%RTESD
%MPES
%TSM
%MUTEX

Ce type de macro-instructions génèrent les macro-instructions %OPTION, %NTACHES et %SUMAX appropriées pour chaque système.

Ces macro-instructions peuvent figurer dans n’importe quel ordre. Elles sont toutes facultatives. Tout autre macro-instruction employée dans cette étape sera tuée par MACP.

Cette phase se termine à la rencontre de la première macro-instruction %NIVEAU.

4.2.3 - Configuration

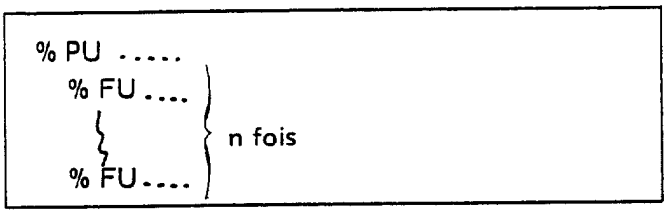
Dans cette phase, l’utilisateur décrit toutes les unités physiques de son système, toutes les unités fonctionnelles qu’il utilise, et les liaisons unités physiques - unités fonctionnelles.
C’est dans cette phase que seront générées les tables d’unités physiques.

Cette description se fait par niveau : chaque niveau est déclaré par la macroinstruction
%NIVEAU 999 KSTOR = [999]
A chaque niveau peuvent correspondre plusieurs sous-niveaux. Chaque unité physique ou chaque coupleur multiplexé est rattaché à un sous-niveau. Pour chaque niveau on décrira donc tous les sous-niveaux occupés.
Apres chaque description d’unité physique (% PU) on décrira les-unités fonctionnelles rattachées à cette unité.
Après chaque description de coupleur multiplexé occupant le niveau i, on décrira le(s) périphérique(s) connecté(s) à ce coupleur par des macroinstructions %PUMPX, et pour chaque périphérique connecté les unités fonctionnelles qui lui sont associées (par %FU).

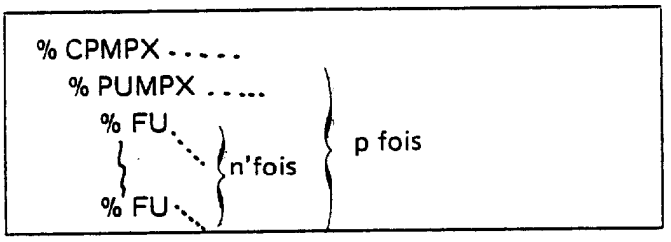
Pour décrire les périphériques dits "standard", on peut utiliser les macroinstructions disponibles dans la bibliothèque : %TTY, %HR, %HP, %CR, %LP, %MT . . .
Pour décrire les périphériques reliés à un coupleur de type MXP 04 on dispose de macroinstructions particulières %CPMUX4 et %PUMUX4.
Pour la description des macroinstructions, se reporter aux pages suivantes.
Cette phase se termine par la macroinstruction %ENDGEN.

On aura l'organisation suivante :

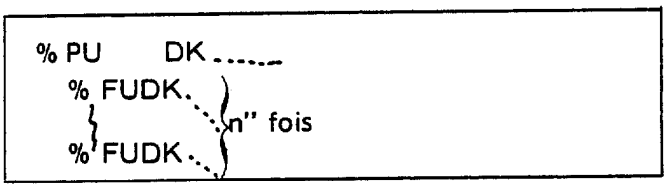
%NIVEAU i



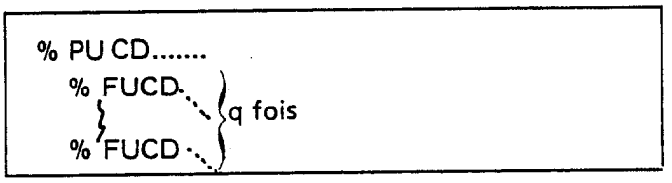
Description d'un périphérique non standard et non multiplexé



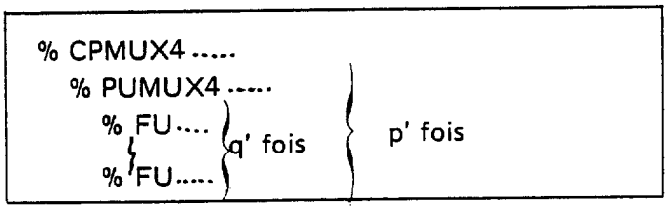
Description d'un coupleur multiplexé et des unités physiques qui s'y rattachent



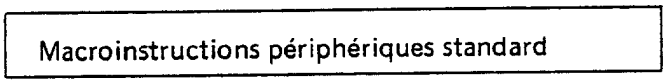
Description d'une unité physique disque à tête fixe



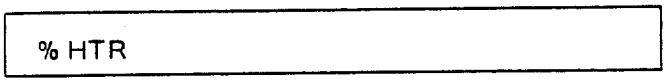
Description d'une unité physique disque à cartouche



Description d'un coupleur MXP04 et des périphériques qui s'y rattachent



Seulement en niveau 14 ou 15



Description horloge temps réel (1 seule fois par configuration)

NB : i peut varier de 1 à 15

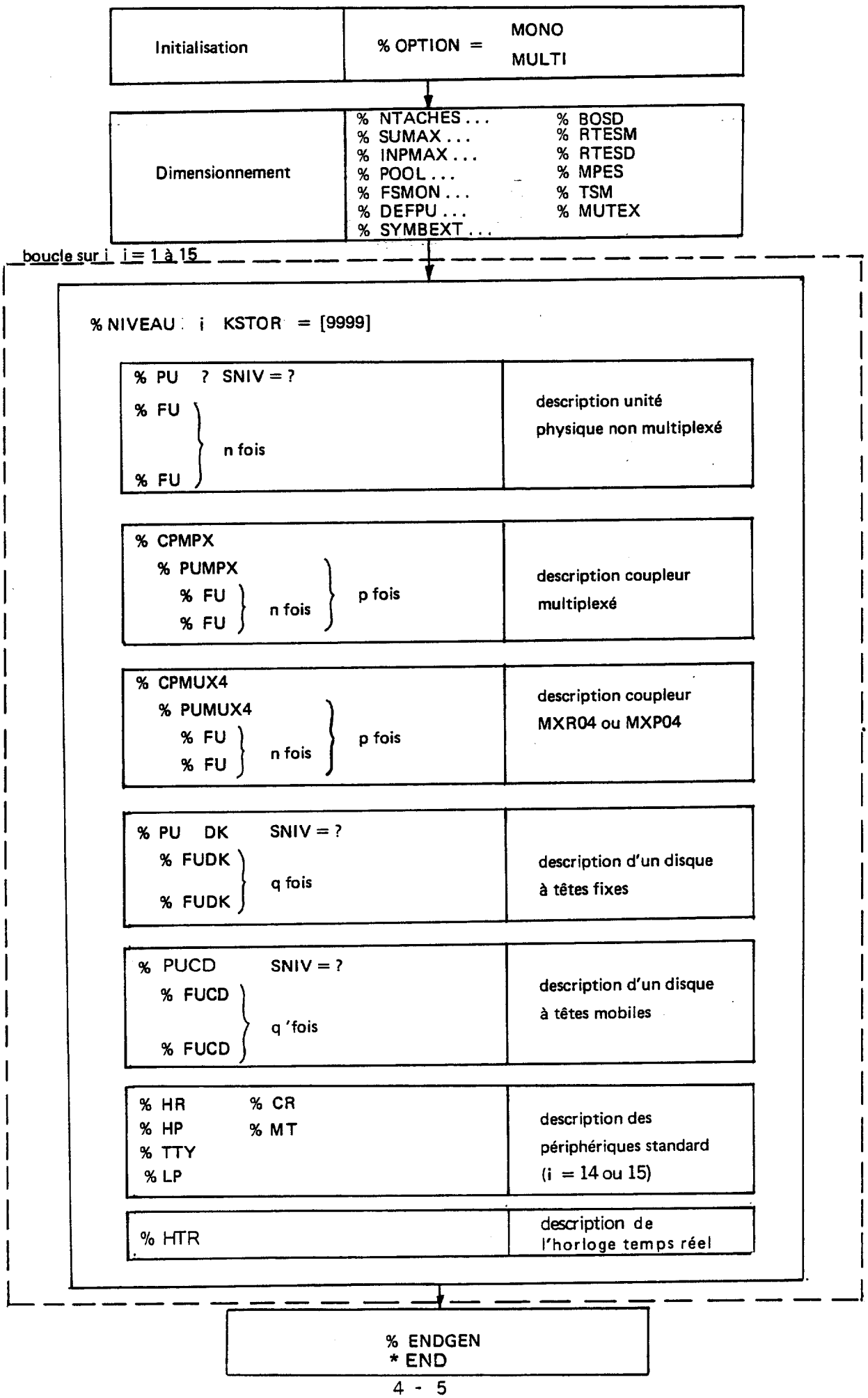
Chaque rectangle correspond à un ou plusieurs sous niveaux. On peut avoir plusieurs rectangles identiques à l'intérieur d'un même niveau. Chaque rectangle peut être omis.
Il est inutile de définir un niveau sur lequel aucun périphérique n'est connecté.

Remarques :

La description d'un niveau doit être complètement terminée avant la prochaine macroinstruction %NIVEAU.
ex : cette configuration n'est pas autorisée

```
% NIVEAU i
   |
   |
% NIVEAU j
   |
   |
% NIVEAU i
```

On peut décrire les niveaux dans n'importe quel ordre


Initialisation
% OPTION = MONO MULTI
Dimensionnement
% NTACHES . . .
% SUMAX . . .
% INPMAX . . .
% POOL . . .
% FSMON . . .
% DEFPU . . .
% SYMBEXT . . .
% BOSD
% RTESM
% RTESD
% MPES
% TSM
% MUTEX
boucle sur i i = 1 à 15
% NIVEAU i KSTOR = [9999]
% PU ? SNIV = ?
% FU
% FU
n fois
description unité physique non multiplexé
% CPMPX
% PUMPX
% FU
% FU
n fois
p fois
description coupleur multiplexé
% CPMUX4
% PUMUX4
% FU
% FU
n fois
p fois
description coupleur MXR04 ou MXP04
% PU DK SNIV = ?
% FUDK
% FUDK
q fois
description d'un disque à têtes fixes
% PUCD SNIV = ?
% FUCD
% FUCD
q 'fois
description d'un disque à têtes mobiles
% HR
% HP
% TTY
% LP
% CR
% MT
description des périphériques standard (i = 14 ou 15)
% HTR
description de l'horloge temps réel
% ENDGEN
* END

4.3 - DESCRIPTION DE LA BIBLIOTHEQUE

4.3.1 - Phase initialisation

```
%OPTION = ?
```

Cette macroinstruction permet à l'utilisateur de préciser la version du noyau d'IOCS utilisé, c'est-à-dire soit la version mono-tâche, soit une version multi-tâches (pour les calculateurs ayant l'option scheduler microprogrammé MTS 16).

Syntaxe :

```
% ØPTIØN = { MØNØ }
          { MULTI }
```

Erreurs détectées :

L'erreur "paramètre incorrect'* sera détectée par GENIO si le paramètre est différent de MONO et MULTI.

Cette macroinstruction sera obligatoirement présente et sera la première du jeu d'instructions

4.3.2 - Phase dimensionnement

```
%NTACHES = ?
```

Cette macroinstruction permet d'indiquer le nombre maximum de tâches susceptibles de travailler avec IOCS (dans le cas de l'option multi-tâches).
Elle n'est utile que lorsque l'on a %OPTION = MULTI, elle sera donc interdite en version mono-tâche.

Syntaxe :

```
%NTACHES = 999
```

Erreurs détectées :

L'erreur "PARAMETRE INCORRECT" est détectée si :

- le paramètre n'est pas de type décimal
- le paramètre est supérieur à 128.

L'erreur "MACROINSTRUCTION INTERDITE" est détectée si cette macroinstruction est utilisée pour une version mono-tâche.

Option prise par défaut :

Cette macroinstruction peut être omise, l'option prise par défaut sera alors :

```
%NTACHES = 16
```

%SUMAX = ?

Cette macroinstruction permet à l'utilisateur d'indiquer le nombre maximum d'unités symboliques de son système.

Syntaxe :

%SUMAX = 999

Erreurs détectées :

L'erreur "paramètre incorrect" est détectée si :

- le paramètre n'est pas de type décimal
- le paramètre est supérieur à 128.

Option prise par défaut :

Si cette macroinstruction est omise, on prendra par défaut :

%SUMAX = 16

%INPMAX = ?

Cette macroinstruction permet à l'utilisateur de fixer le nombre maximum de caractères qui seront échangés lors d'un échange avec code d'arrêt en cas de non concordance avec les codes de la table des codes d'arrêt.

Syntaxe :

%INPMAX = 9999

Erreurs détectées :

L'erreur "paramètre incorrect" est détectée dans l'un des cas suivants :

- le paramètre n'est pas de type décimal
- le paramètre est supérieur à 4096.

Options prises par défaut :

On prendra par défaut 80 caractères si cette macroinstruction est omise, soit la macroinstruction

%INPMAX = 80

%PØØ NBUF = ? LBUF = ?

Cette macroinstruction permet de décrire “le pool buffer” en fixant le nombre de buffers du pool d’une part (NBUF = ?) et la longueur en mots de chaque buffer d’autre part (LBUF = ?).

Remarque :

Le nombre de buffers est limité à 31.

Syntaxe :

%POOL NBUF = 99 LBUF = 9999

Erreurs détectées :

L’erreur “paramètre incorrect” est détectée dans l’un des cas suivants :

- Pour le paramètre NBUF :
. le paramètre n’est pas de type décimal
. le paramètre est supérieur à 31.

- Pour le paramètre LBUF :
. le paramètre n’est pas de type décimal
. le paramètre est supérieur à 4096.

Option prise par défaut :

Si cette macroinstruction est omise, on prendra par défaut un nombre de buffers égal à 1 et une longueur de 80 mots, c’est-à-dire %POOL NBUF = 1 LBUF = 80

```
%FSMØN  ?    NUM  = ?
```

Cette macroinstruction permet de définir une fonction spéciale moniteur non standard.

Signification des paramètres :

a) Code mnémonique de la fonction spéciale :

Le premier paramètre est le code de la fonction spéciale. C'est une chaîne de type symbole de 5 caractères au maximum.

Cette chaîne doit être unique car, si XXXXX est le code mnémonique d'une fonction spéciale, GENIO associe à cette fonction un module dont le point d'entrée a pour nom symbolique FXXXXX dans la table TBFSM.

Ce module devra être assemblé isolément puis relié par édition de liens à l'IOCS généré.

Les codes affectés aux fonctions spéciales standard sont interdits, ils sont connus implicitement par GENIO.

b) Numéro de la fonction (NUM = ?)

C'est un nombre décimal qui indique le numéro attribué à la fonction spéciale définie. Il doit être différent des numéros attribués aux fonctions spéciales standard, c'est-à-dire supérieur ou égal à 2 si on travaille en version mono-tâche, ou supérieur ou égal à 8 en version multi-tâches.

Remarque :

Si l'utilisateur donne pour nom à une fonction spéciale moniteur un nom déjà attribué à une fonction spéciale standard, aucune erreur ne sera détectée par GENIO, mais il y aura une erreur lors de l'assemblage (double définition d'étiquettes). Il lui faudra donc recommencer la génération.

Fonctions spéciales moniteur standard :

Les fonctions standard sont incluses dans l'IOCS généré sans qu'il soit besoin de les définir par une macroinstruction FSMON. On trouvera dans le manuel de référence IOCS la description de ces fonctions.

Les fonctions standard sont :

FCLEAR	**num = 0**	
FPUSI	**num = 1**	
FPUA	**num = 2**	**en version multi-tâches seulement**
FPUD	**num = 3**	
FKILL	**num = 6**	
FCLSEL	**num = 7**	

Les fonctions "gestion de volume" sont :

DMONT	**num = 8**
MONT	**num = 9**
READ	**num = 10**

Syntaxe :

%FSMON XXXXX NUM = 999

Erreurs détectées :

L'erreur "paramètre incorrect" est détectée dans l'un des cas suivants :

- Pour le premier paramètre (nom de FSMON) :
. ce paramètre n'est pas de type symbole
. ce paramètre a plus de 5 caractères

- Pour le paramètre NUM = ?
. ce paramètre n'est pas de type décimal
. il est supérieur à 128
. il est inférieur à 2 si version mono-tâche ou inférieur à 8 si version multi-tâches.

```
%DEFPU ?    CØDE = ?    LZØNE = ?    SP = ?
```

Cette macroinstruction permet de définir un type de périphérique non standard.

Signification des paramètres :

a) Type de périphériques

On sait que les fonctions particulières à un périphérique (échanges, fonctions de positionnement) sont réalisées par le driver correspondant à ce périphérique.
Plusieurs périphériques peuvent utiliser le même driver à condition d'être de même type.
On associe donc à chaque type de périphérique un driver chargé d'exécuter les fonctions qui lui sont propres.
Le type de périphérique est représenté par une chaîne de type symbole de trois caractères au maximum.
Lors de la génération, GENIO associe automatiquement à un périphérique de type XXX un driver tel que :

- son point d'entrée ait pour nom symbolique DRVXXX
- l'adresse sur laquelle pointe la base L dans son LOCAL ait pour nom symbolique LOCXXX.

Lorsqu'un utilisateur veut introduire dans son installation un périphérique de type particulier, il doit écrire le driver correspondant et le définir par une macro-instruction DEFPU afin que GENIO puisse le reconnaître dans la suite de la génération.

b) Longueur des codes transférés (CODE = ?)

Ce paramètre précise si le type de périphérique transfère des codes de 8 bits (lecteur de ruban par exemple) ou de plus de 8 bits (lecteur de cartes par exemple).

On notera :

CODE = 8	**pour les périphériques à code 8 bits**
CODE = 16	**pour les périphériques à code supérieur à 8 bits.**

c) Longueur de la zone de travail (LZØNE = ?)

On sait qu'un driver trouve les informations nécessaires à un échange donné dans la table unité physique du périphérique sur lequel est effectué l'échange.
La table unité physique est décrite dans ce manuel.
Il peut arriver qu'un driver ait besoin "d'agrandir" la table unité physique de façon à pouvoir disposer d'une zone de travail (cas du driver disque à cartouche par exemple). Le paramètre LZONE = ? précise la longueur de cette zone de travail. (Elle est limitée à 128 mots).
Cette zone de travail est générée au-dessus de la table unité physique (TUP) standard.

d) Spécification "microprogrammé" (SP = ?)

Certains périphériques peuvent être microprogrammé de manière spécifique. Ce paramètre vaut Y si le périphérique est microprogrammé et N sinon. Ce paramètre peut être omis, l'option prise par défaut est N.

Les périphériques gérés par les drivers contenus dans les bibliothèques BIBDRV - : s (driver standard) et DRVTRA - : s (driver de transmission) sont reconnus implicitement par GENIO.

DRIVERS STANDARD		PARAMETRES GENERES		
DRV	Périphérique/coupleur	CODE =	LZONE =	SP =
HR	Lecteur de ruban perforé rapide	8	0	N
HP	Perforateur rapide	8	0	N
CR	Lecteur de carte	8	0	N
LP	Imprimante rapide	8	0	N
FDD	Disque souple 2F 2D	16	52	N
VM	Disque à cartouche 10 et 20 mb	16	26	N
DK	Disque à têtes fixes	16	0	N
DP	Disque moyenne capacité (50 méga)	16	15	N
TTY	Télétype	8	0	N
SAS	Disque WINCHESTER	16	68	N
FDO	Disque souple SEMS	16	30	N
FD1	Disque souple IBM	16	11	N
IB	Disque souple IBM “MULTI-FU”	16	21	N
HTR	Horloge temps réel	16	0	N
HT2	Imprimante diablo HT2	8	11	N
MT	Bande magnétique	16	3	N
P16	Coupleur MXP16, MXP08	8	0	N
VT	ASYN1V, MUX4, MUX8, MUX16	8	3	N
SMD	Disque à interface SMD	16	64	N
ASM	Coupleur ASYN1V, MUX4	8	10	N
DLC	Coupleur de ligne HDLC	8	10	N
EDC	Coupleurs synchrones	8	16	Y
BDG	Terminaux lecteur de badges	8	36	N
LX	Imprimante Logabax série LX 180	8	11	N
LXM	Imprimante Logabax série LX 180 modem	8	12	N
MC1	MUX4 canal standard	8	21	N
MC2	MUX4 canal gestion DATA-MEDIA TV-6040	8	21	N
TLX	Coupleur de ligne Télex	8	26	N

Les paramètres indiqués ci-dessus peuvent être modifiés par la macro %DEFPU.

Syntaxe :

```
% DEFPU   XXX   CODE { 8 | 16 }   LZØNE = 999   SP = [ { Y | N } ]
```

Erreurs détectées

L'erreur "paramètre incorrect" est détectée dans l'un des cas suivants :

- Pour le paramètre "type" :
. le paramètre n'est pas de type symbole
. sa longueur est supérieure à 3.

- Pour le paramètre CODE :
. il est différent de 8 et de 16

- Pour le paramètre LZONE :
. il n'est pas de type décimal
. il est supérieur à 128

- Pour le paramètre SP :
. il est autre que Y ou N

Remarques :

Si le type de périphérique est l'un de ceux connus implicitement par GENIO, celui-ci ne détectera pas d'erreur lors de la génération, mais il y aura une erreur d'assemblage (double définition d'étiquettes).

`% SYMBEXT ?`

Cette macro permet d 'indiquer le nom d'un module externe utilisé dans une ou plusieurs macroinstructions (en particulier %TUP...). Ce module devra être relié par édition de liens avec le noyau d'lOCS.

Syntaxe :

```
%SYMBEXT   XXXXXX
```

Exemple :

```
%SYMBEXT   TRANSC
```

4.3.3 - Les macros "standard" système

Ces macro-instructions servent à définir des systèmes conventionnels dits "standard". L'utilisateur peut simplifier la phase d'initialisation et de dimensionnement en utilisant les macro-instructions suivantes :

`% BOSD`

`% RTESM`

`% RTESD`

`% MPES`

`% TSM`

`% MUTEX`

Ces macro-instructions regroupent les macro-instructions %OPTION, %NTACHES, %SUMAX, %POOL dont les sont données dans le tableau ci-dessous :

	%TSM	%RTESM	%RTESD	%MUTEX	%BOSD	%MPES
OPTION =	MULTI	MULTI	MULTI	MULTI	MULTI	MULTI
NTACHES =	128	64	128	128	48	128
SUMAX =	0	30	30	30	29	1
POOL NBUF=	0	*	*	*	*	*
POOL LBUF=	0					

* = non générée pour le système

4.3.4 - Phase configuration

```
%NIVEAU ?     [KSTØR = ?]
```

Cette macroinstruction indique le début de la description d'un niveau d'interruption : toutes les macroinstructions qui suivent jusqu'à la prochaine macroinstruction % NIVEAU concernent les périphériques reliés aux différents sous-niveaux de ce niveau.

Signification des paramètres :

— le paramètre NIVEAU précise le numéro de niveau. C'est un nombre compris entre 1 et 15.
— le paramètre taille (KSTOR = ?) donne la taille en mots de la pile K de la tâche gérant le niveau. Ce paramètre doit être inférieur à 512.

Si le paramètre KSTOR est omis, on prendra par défaut la taille standard suffisante pour l'utilisation des drivers standards (30 mots).

Syntaxe :

```
% NIVEAU 99      KSTOR = [9999]
```

Erreurs détectées :

L'erreur "paramètre incorrect" sera détectée par GENIO si :

— l'un des deux paramètres n'est pas de type décimal
— le premier paramètre (niveau) est supérieur à 15
— le deuxième paramètre (KSTOR) est supérieur à 512.

```
%PU  ?   SNIV = ?   MODE = ?   ADR = ?   ITN = ?   CONNEX = ?   IØP = ?
```

Cette macroinstruction définit une unité physique non multiplexée connectée sur le niveau précédemment décrit.
Elie est obligatoirement suivie par une ou plusieurs macroinstructions % FU (description des unités fonctionnelles associées à cette unité physique).

Signification des paramètres :

- Paramètre "nom de pu" :
Le premier paramètre décrit le type de périphérique. Si celui-ci n'est pas standard, il doit obligatoirement être décrit auparavant par une macroinstruction DEFPU.

- Paramètre "sous niveau" (SNIV = ?)
Ce paramètre indique le sous-niveau sous lequel est connecté l'unité physique. Il est fonction du mode de fonctionnement. Il doit être omis si le mode est PS, inférieur à 16 si le mode est PP, et inférieur à 48 dans les autres cas.

- Paramètre "mode" (MODE = ?)
Ce paramètre indique le mode de fonctionnement du périphérique. Il vaut :

. PS	**si programmé simple**
. PP	**si programmé prioritaire**
. LDC	**(Low speed Data Channel)**
. MDC	**(Médium speedDataChannel)**
. HDC	**(High speed Data Channel)**

- Paramètre "adresse" (ADR = ?)
Il s'agit de l'adresse du coupleur du périphérique. C'est un nombre hexadécimal.

- Paramètre "itn" (ITN= ?)
Il s'agit du niveau des interruptions canal. Ce paramètre doit être omis si le mode est PP ou PS, il est obligatoire dans les autres modes.
Si le mode est HDC, "itn" aura une valeur comprise entre 0 et 7
Si le mode est MDC, "itn" sera inférieur à 16
Si le mode est LDC, "itn" sera inférieur à 64.

- Paramètre "connexion" (CONNEX = ?)
Il précise le mode de connexion (c'est-à-dire le type de coupleur sur lequel est connecté le périphérique). Il peut être :

. STD Standard

. ASYV1 Asynchrone une voie

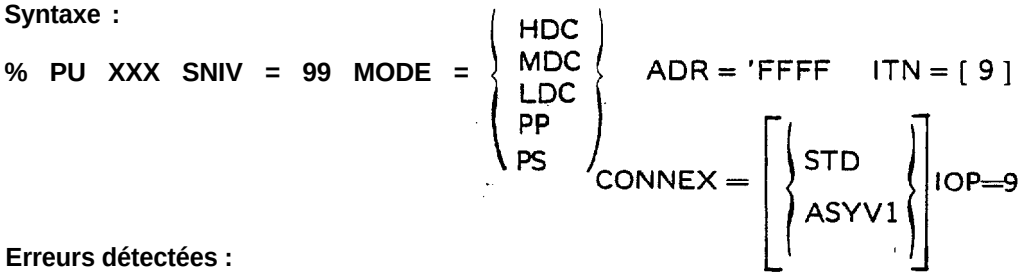
Il peut être omis, l'option prise par défaut est STD.
Les périphériques reliés par un coupleur de type MXP04 font l'objet de macroinstructions spéciales

- Paramètre "IOP" (IOP = ?)

Ce paramètre précise le numéro du processeur qui gérera les entrées-sorties. Ce paramètre est interdit si le mode est PP.

Ce paramètre vaut 0, 1, 2, 3. S'il est omis, le processeur de calcul (CPU n° 0) se chargera de gérer les entrées-sorties.

Syntaxe :



Erreurs détectées :

L'erreur "paramètre incorrect" est détectée si :

- Pour le paramètre "nom pu" :
  . il n'est pas de type symbole
  . il a plus de 3 caractères

- Pour le paramètre "sous-niveau" :
  . si le mode est PP, il est supérieur à 16
  . si le mode est différent de PP il est supérieur à 48.

- Pour le paramètre ""mode"
  . il est différent de PP, HDC, MDC, LDC ou PS.

- Pour le paramètre "adresse"
  . ce paramètre n'est pas de type hexadécimal.

- Pour le paramètre "ITN"
  . il est présent à tort car le mode est PP ou PS
  . il est supérieur à 7 si le mode est HDC
  . il est supérieur à 15 si le mode est MDC
  . il est supérieur à 63 si le mode est LDC.

- Pour le paramètre "connex"
  . il est différent de STD et ASYV1

- Pour le paramètre "IOP"
  . il est supérieur à 3
  . il est présent si le mode est PP ou PS.

Remarques :

- l'utilisation d'un type de périphérique non défini ne donnera pas lieu à une détection d'erreur lors de la génération et ne pourra donc pas être corrigée à ce moment là. Cette erreur sera détectée à l'assemblage (symboles non définis). Ce type d'erreur est fatal et nécessite de recommencer la génération.

%TUP - ? = ?

Cette macroinstruction permet d'initialiser l'extension de la dernière TUP générée, au moyen d'un nombre ou d'un symbole externe défini par %SYMBEXT. Elle peut suivre une macro-%PU, %PUMUX4, %PUMPX, %PUDX.

Le premier paramètre donne le déplacement par rapport au début de la TUP.
Le deuxième donne la valeur ou le symbole à y charger. Attention, aucune vérification n'est faite par rapport au paramètre LZONE de la macro %DEFPU qui définit la longueur de l'extension de TUP.

%FU ? CDE= ? SENS= ? FINDIC = ?

Cette macroinstruction définit une unité fonctionnelle (autre que disque) associée à la dernière unité physique (PU, PUMPX, PUMUX4) déclarée.

Signification des paramètres :

- Paramètre "FU" :
. il précise le numéro de FU et peut être exprimé en décimal (< 126) ou par sa clé symbolique (TS, LP, etc. . .).

- Paramètre "Cde" : (CDE = ?)
. ce paramètre est le mot de commande. Il est de type hexadécimal.

- Paramètre "sens" (SENS = ?)
Ce paramètre indique le sens des échanges qui seront effectués sur cette unité fonctionnelle.
Il peut être égal à :
. I si elle n'effectue que des entrées
. O si elle n'effectue que des sorties
. IO si elle effectue des entrées et des sorties.

- Paramètre "Findic" (FINDIC = ?)
. ce paramètre vaut 0 ou 1, sa signification dépend du type de périphérique sur lequel pointe la FU

Syntaxe :

% FU { 999 / XX } CDE = 'FFFF SENS = { I / Ø / IØ } FINDIC = { 0 / 1 }

Erreurs détectées :

- Pour le paramètre "FU" :

L'erreur "paramètre incorrect" sera détectée si :

. il est de type décimal supérieur à 125,
. le symbole n'est pas reconnu (voir § 2.6.1 du M.R. BOSD).

- Pour le mot de commande "cde"
L'erreur "paramètre incorrect" sera détectée s'il n'est pas de type hexadécimal.

- Pour le sens :
L'erreur "paramètre incorrect" sera détectée si il est autre que I, O ou IO

- Pour le paramètre "Findic"
L'erreur "paramètre incorrect" sera détectée s'il est différent de 0 ou 1.

Remarque :

Deux unités fonctionnelles ne peuvent avoir le même numéro, l'erreur n'est pas détectée au moment de la génération mais au moment de l'assemblage. Cette erreur est fatale et nécessite de recommencer la génération.

```
%CPMPX ? SNIV = ?     MØDE = ?    ADR = ?   ITN = ?   CONNEX = ?
                                              IOP = ?   NBV = ?
```

Cette macroinstruction permet de définir un coupleur multiplexé. Pour chaque coupleur multiplexé, l'utilisateur devra écrire une macroinstruction CPMPX. S'il n'est pas connu implicitement par GENIO le coupleur doit avoir été décrit par une macroinstruction DEFPU.

Cette macroinstruction est obligatoirement suivie par une ou plusieurs macroinstructions PUMPX.

Signification des paramètres :

Les sept premiers paramètres ont la même signification que ceux de la macroinstruction %PU

Paramètre NBV :

Ce paramètre précise le nombre de voies utilisées sur le coupleur que l'on définit. Ce nombre est inférieur ou égal à 16.

Syntaxe :

% CPMPX XXX SNIV = 99 MODE = { HDC | MDC | LDC | PP } ADR = 'FFFF ITN = [99]

CONNEX = [{ MUX16 | MUX8 }] IOP = 9 NBV = 99

Erreurs détectées :

- **pour les sept premiers paramètres, les erreurs sont les mêmes que dans la macroinstruction %PU**
- **pour le paramètre NBV ; l'erreur "paramètre incorrect" sera détectée si :**
  - **. il n'est pas de type décimal**
  - **. il est supérieur à 16.**

Remarques :

- **Les coupleurs disque, floppy disque, ainsi que les Coupleurs de type MXP04 ne peuvent être décrits par cette macroinstruction.**
- **l'utilisation d'un coupleur non défini ne donnera pas lieu à une détection d'erreur lors de la génération. L'erreur ne sera détectée qu'au moment de l'assemblage (symboles non définis), et nécessitera de recommencer la génération.**

%PUMPX ? VOIE= ?

Cette macroinstruction définit un périphérique connecté sur le coupleur multiplexé CPMPX décrit dans le paragraphe précédent.

Signification des paramètres :

- Paramètre "typu" :
Ce paramètre précise le type de périphérique. C'est une chaîne de 3 caractères alphanumériques au maximum.
S'il n'est pas l'un de ceux reconnus implicitement par GENIO, il doit obligatoirement avoir été l'objet d'une macroinstruction DEFPU.

- Paramètre "voie" (VOIE = ?)
Ce paramètre précise sur quelle voie s'effectue l'échange. Il doit être inférieur au nombre de voies déclaré dans la macroinstruction CPMPX.

Syntaxe :

%PUMPX XXX VOIE = 99

Erreurs détectées :

L'erreur "paramètre incorrect" est détectée si :

- Pour le paramètre "typu" :
. il n'est pas de type symbole
. il a plus de trois caractères

- Pour le paramètre "voie" :
. il n'est pas de type décimal
. il est supérieur au nombre de voies du coupleur.

L'erreur "macroinstruction interdite" est détectée si %PUMPX n'a pas été précédée de %CPMPX.

Remarque :

- l'initialisation d'un type de périphérique non défini ne donnera pas lieu à une erreur lors de la génération. L'erreur ne sera détectée qu'au moment de l'assemblage et nécessitera de recommencer la génération.

%PUDX ? BEGIN = ? NUMBER = ? STEP = ? ADR = ?

Cette macro décrit un coupleur démultiplexé, c'est à dire un coupleur connu comme une seule unité physique, mais utilisant plusieurs sous niveaux d'interruption égaux à :

BEGIN
BEGIN + STeP } **NUMBER fois**
BEGIN + STEP + STEP

%PUDX doit être suivi d'une ou plusieurs macros %FU.

Signification des paramètres :

- Paramètre "BEGIN" :

C'est la valeur du sous-niveau de plus petit rang que l'utilisateur veut gérer.

- Paramètre "NUMBER" :

C'est le nombre de sous-niveau que l'utilisateur désire gérer.

- Paramètre "STEP"

C'est la différence existant entre le sous niveau de rang j et le sous-niveau de rang i. i est le rang du premier sous-niveau géré et j le rang du second.

%CPMUX4 MØDE = ? ADR = ? IØP = ? [CONNEX = { MUX 4 / ASYV 1 / STD }]

Cette macroinstruction permet de définir un coupleur du type MXP 04. Elle est obligatoirement suivie de une ou plusieurs macroinstructions PUMUX4.

Signification des paramètres

- Paramètre "mode"

Ce paramètre indique le mode de fonctionnement des périphériques connectés sur le coupleur

- **PP si programmé prioritaire**
- **LDC si Low Data Channel**
- **MDC si Medium Data Channel**
- **HDC si High Data Channel**

- Paramètre "adresse" (ADR = ?)

C'est un nombre hexadécimal qui précise l'adresse du coupleur.

- Paramètre IOP

C'est un nombre décimal inférieur à 4 qui précise le numéro du processeur qui gère les entrées-sorties.

Ce paramètre doit être omis si MODE = PP. S'il est omis avec MODE = { MDC / LDC / HDC }, on prendra par défaut la valeur 0 (processeur de calcul).

- Paramètre CONNEX = il précise le mode de connexion (c'est-àdire le type du coupleur) ; il peut être MUX4, ASYV1, STD. Par défaut CONNEX = MUX4.

Syntaxe :

%CPMUX4 MODE = { PP / LDC / MDC / HDC } ADR = 'FFFF IOP = [91] [CONNEX = [XXXX]]

Erreurs détectées :

L'erreur "paramètre incorrect" sera détectée si :

- **le paramètre MODE est autre que PP, LDC, MDC ou HDC**
- **le paramètre ADR n'est pas de type hexadécimal**
- **le paramètre IØP est supérieur à 3**
- **le paramètre CONNEX est différent de MUX4, ASYV1, STD.**

Remarque : Un coupleur de type MXP04 est un coupleur dont le sous-niveau d'entrée et le sous-niveau de sortie d'une même voie sont différents.

%PUMUX4 ? SNIV1 = ? SNIV2 = ? ITN = ? VØIE = ?

Cette macroinstruction définit une unité physique connectée sur un coupleur de type MXP04. Elle doit être précédée de la macro CPMUX4.

Remarque préliminaire :

Si le périphérique peut fonctionner en entrée et sortie, les sous niveaux dès interruptions entrée et sortie différents.

- pour un fonctionnement en half duplex, on considérera le périphérique comme une seule unité physique : donc une seule macro PUMUX4 précisant le sous-niveau entrée et le sous-niveau entrée et le sous-niveau sortie.

- pour un fonctionnement en full duplex, on considérera le périphérique comme 2 unités physiques : donc 2 macros PUMUX4, l'une en entrée, l'autre en sortie. La déclaration de la ou des FU associée(s) (%FU) doit être faite après la macro %PUMUX4.
Deux macro %PUMUX4 doivent donc être séparées par une ou des macro(s) %FU ; seule la macro %FU peut être placée entre les macros %PUMUX4 associées à une même macro %CPMUX4.

Signification des paramètres

- Paramètre "nompu" (PUMUX4?)
Ce paramètre précise le type de périphérique. S'il n'est pas connu implicitement par GENIO, il doit avoir été défini auparavant par une macro-instruction DEFPU.

- Paramètre SNIV1
Ce paramètre est obligatoire. Il précise le sous-niveau (réception si fonctionnement en half-duplex) des interruptions du périphérique.
C'est un nombre décimal inférieur à 16 si le mode de fonctionnement (défini dans CPMUX4) est "programmé prioritaire", ou HDC et MDC, inférieur à 48 en mode LDC.

- Paramètre SNIV2
Le paramètre SNIV2 n'est utile que dans le cas d'un périphérique en entrée-sortie fonctionnant en half-duplex. Il indique alors le sous-niveau réception des interruptions du périphérique. Il doit être omis dans tous les autres cas. Il suit les mêmes règles que SNIV1 mais doit être supérieur de celui-ci et tel que SNIV2>SNIV1.

- Paramètre ITN
Ce paramètre est un nombre décimal qui précise le niveau des interruptions canal. Il est inférieur à 64 en LDC, 8 en HDC, 16 en MDC.
Il est obligatoire en mode LDC, MDC, HDC (paramètre donné dans la macro CPMUX4) et interdit en mode PP.

- Paramètre VOIE
C'est un nombre décimal inférieur à 4 qui précise sur quelle voie du coupleur est connecté le périphérique.

Syntaxe :

%PUMUX4 XXX SNIV1 = 99 SNIV2 = [99] ITN = [99] VOIE = { 0 | 1 | 2 | 3 }

Erreurs détectées :

- **"macro interdite" si elle n'est pas précédée de %CPMUX4**
- **"paramètre incorrect"**
  - **. le type de périphérique n'est pas de type symbole, on a plus de 3 caractères.**
  - **. SNIV1 ou SNIV2 est**
    supérieur à 15 en mode PP, HDC et MDC
    ou supérieur à 47 en mode LDC
  - **. ITN est supérieur à 63**
  - **. VOIE est supérieur à 3**
- **"paramètre interdit"**
  - **. ITN est présent alors que le MODE est PP.**

Remarque :

- **l'utilisation d'un type de périphérique non défini provoquera une erreur d'assemblage. Il faudra recommencer la génération.**

%PUCD SNIV = ? MØDE = ? ADR = ? ITN = ? IØP = ? NBV = ?

Notations :

On appelle unité d'échange multiplexée, l'ensemble des unités gérés par le même coupleur.

Cette macroinstruction permet de décrire une unité d'échange multiplexée disque 10 ou 20 mb. Pour chaque unité d'échange multiplexée présente dans son installation, l'utilisateur devra décrire une macroinstruction %PUCD.

Signification des paramètres :

- Paramètre "sniv" (SNIV = ?)
Ce paramètre indique le sous-niveau d'interruption sur lequel est connecté l'unité d'échange multiplexée.

- Paramètre "mode" (MØDE = ?)
Ce paramètre précise le mode de connexion. Ce ne peut être que HDC ou MDC.

- Paramètre "adresse" (ADR = ?)
Il indique l'adresse du coupleur. C'est un nombre hexadécimal.

- Paramètre "itn" (ITN = ?)
Ce paramètre est obligatoire. Il précise le numéro d'interruption canal.

- Paramètre "iop" (IØP = ?)
Le cinquième paramètre précise le processeur d'entréesortie qui gère le canal. Il peut varier entre 0 et 3. S'il est omis, par défaut, le processeur de calcul servira à gérer les entrées-sorties.

- Paramètre "nbv" (NBV = ?)
Ce paramètre précise le nombre de voies. Il est inférieur à 5.

Syntaxe :

%PUCD SNIV = 99 MØDE = { HDC / MDC } ADR = "FFFF ITN = 99 IØP= 9 NBY = 99

Erreurs détectées :

Les erreurs détectées sur les paramètres SNIV, ADR, ITN, IØP sont les mêmes que dans la macroinstruction PU.

- Pour le paramètre MODE :
L'erreur "paramètre incorrect" est détectée si le paramètre est différent de HDC ou MDC.

- Pour le paramètre "VOIE" :
L'erreur "paramètre incorrect" sera détectée si :
. le paramètre n'est pas décimal
. le paramètre est supérieur ou égal à 5.

```
%FUCD ? VOIE = ? ADRCYL = ? NBCYL = ? FIXE = ?
```

Cette macroinstruction permet la description d'une unité fonctionnelle disque à cartouche appartenant à l'un des disques de l'unité d'échange multiplexée définie précédemment par une macroinstruction PUCD OU PUVM.

N.B. : Un cylindre contient 48 secteurs, c'est à dire 6 K mots.

Signification des paramètres :

- Paramètre "numfu"

Ce paramètre indique le numéro de FU et peut être exprimé en décimal (< 126) ou en symbolique (D1, E1, etc. . .).

- Paramètre "num voie" :

Ce paramètre précise le numéro de voie sur laquelle est connectée la FU. Ce nombre peut prendre des valeurs de 0 à 3. Il doit être inférieur au nombre de voies défini dans la macroinstruction PUCD.

- Paramètre "début"

Ce paramètre précise le numéro de cylindre où commence la FU disque. Il doit être inférieur à 400.

- Paramètre "nombre cylindre" :

Ce paramètre indique la longueur de la FU disque en nombre de cylindres. Ce paramètre doit être inférieure à 401. De plus la somme "début" + "nombre cylindre" doit être inférieure à 401.

- Paramètre "fixe" :

Ce paramètre vaut Y si la FU se trouve sur le plateau fixe et N si elle se trouve sur le plateau mobile.

Syntaxe :

% FUCD {999 / XX} VOIE = 9 **ADRCYL = 999 NBCYL = 999** **FIXE =** {Y / N}

Erreurs détectées :

L'erreur "macroinstruction interdite" est détectée si cette macroinstruction n'a pas été précédée d'une macroinstruction PUCD ou PUVM.

L'erreur "paramètre incorrect" est détectée si :

- Pour le paramètre "numfu"

. il est de type décimal et supérieur à 125,
. le symbole n'est pas reconnu (voir § 2.6.1 du M.R. BOSD).

- Pour le paramètre "voie"

. il n'est pas de type décimal
. il est égal ou supérieur au nombre de voies défini dans PUCD

- Pour le paramètre "adresse cylindre" :
  . il n'est pas de type décimal
  . il est supérieur à 400.

- Pour le paramètre "fixe" :
  . il est différent de Y ou de N.

- L'erreur "INCOMPATIBILITE ENTRE ADRCYL ET NBCYL" est détectée si :
  . la somme "nombre de cylindre + adresse cylindre" est supérieure à 400.

Remarques :

Deux unités fonctionnelles ne peuvent avoir le même numéro. Cette erreur n'est détectée qu'au moment de l'assemblage. Elle est fatale et nécessite de regénérer la configuration.

```
%PUDP SNIV = ? MODE = ? ADR = ? ITN = ? IOP = ? NBV =  ?
```

Cette macro décrit une unité d'échange multiplexée "disque de moyenne capacité" qui peut contenir 1 à 4 unités de 50 méga-octets.

Les paramètres, la syntaxe et les erreurs sont les mêmes que dans la macro %PUCD.

```
%FUDP ? VOIE = ? ADRCYL = ? NBCYL = ?
```

Cette macro décrit une unité fonctionnelle "disque de moyenne capacité" appartenant à l'un des disques de l'unité d'échange multiplexée définie précédemment par une macro %PUDP.

NB : Un cylindre contient 480 secteurs, c'est à dire 60 K mots.

Les paramètres ont la même signification que les 3 premiers paramètres de %FUCD.

Remarque : Le nombre de cylindres d'une FU disque moyenne capacité est limité à la taille du support c'est-à-dire 400 cylindres.

```
%PUVM SNIV = ? MODE = ? ADR = ? ITN = ?  IOP = ? NBV = ?
```

Cette macro décrit une unité d'échange multiplexée "disque 20 mb ou disque 10 mb" pouvant contenir 1 à 4 unités de 20 méga-octets.

Les paramètres, la syntaxe et les erreurs sont les mêmes que dans la macro %PUCD.

```
%FUVM ? VOIE = ? ADRCYL = ? NBCYL = ? FIXE = ?
```

Cette macro décrit une unité fonctionnelle "disque 20 mb" appartenant à l'un des disques de l'unité d'échange multiplexée définie précédemment par une macro %PUCD ou %PUVM.

NB : un cylindre de la partie mobile contient 48 secteurs, c'est à dire 6 k mots
un cylindre de la partie fixe contient 144 secteurs, c'est à dire 18 k mots

Les paramètres ont la même signification que pour la macro %FUCD.

Remarque : le nombre de cylindres d'une FU est limité à la taille du support, c'est à dire : 400 cylindres.

```
%PUIB SNIV = ? MODE = ? ADR = ? ITN = ? IOP = ?
```

Cette macro décrit une unité d'échange "floppy disque format IBM multi-FU".

Les paramètres, la syntaxe et les erreurs sont les mêmes que dans la macro %PUCD.

```
%FUIB ?  VOIE = ? SENS = ? FINDIC = ? ADRSEC = ?  NBSEC = ?
```

Cette macro permet la description d'une unité fonctionnelle "floppy disque format IBM multi-FU" appartenant à l'unité d'échange définie précédemment par une macro-instruction %PUIB.

N.B. : un secteur contient 128 mots.

Signification des paramètres :

- Paramètres "numfu" : il indique le numéro de FU et peut être exprimé en décimal ou en symbolique (F1, F2, etc. . .).

- Paramètre "numvoie" : il indique le numéro d'unité sur laquelle est connectée la FU. Ce nombre peut prendre les valeurs 0 à 3.

- Paramètre "sens" :
  I = input
  O = output
  IO = input - output

- Paramètre "findic" : doit être égal à 1 ; il indique que la taille d'un secteur est de 128 mots.

- Paramètre "adresse secteur" : contient l'adresse secteur début de FU. Il doit être inférieur à 962.

- Paramètre "nombre secteur" : il indique la longueur en secteur de la FU. Ce paramètre doit être inférieur à 963. La somme "adresse secteur" + "nombre de secteurs" doit être inférieure à 963.

```
%FUDK ?    ADRDK = ?    NSECT = ?
```

Cette macroinstruction sert à définir une unité fonctionnelle disque à têtes fixes. Elle doit obligatoirement être précédée d'une macroinstruction %PU DK ou d'une autre macroinstruction %FUDK.

Signification des paramètres :

- Paramètre "numfu"
Il représente le numéro de FU (décimal ou symbolique).

- Paramètre "adresse" (ADRDK = ?)
Il représente le numéro de secteur du disque où commence la FU disque. Il doit être inférieur à 8192 (256 pistes de 32 secteurs)

- Paramètre, "nombre de secteurs" (NSECT = ?)
Il précise le nombre de secteurs occupés par la FU disque. Il doit être inférieur à 8193.

- La somme adresse de début de FU + nombre de secteurs doit être inférieure à 8193.

Syntaxe :

```
% FUDK { 999 } ADRDK = 9999 NSECT =
       { XX  }
```

Erreurs détectées :

L'erreur "paramètre incorrect" est détectée si :

- le paramètre "NUMFU" est supérieur à 125 ou, s'il est exprimé en symbolique, n'est pas reconnu (voir § 2.6.1 du M.R. BOSD),
- le numéro de secteur (ADRDK) est supérieur à 8191,
- le nombre de secteur est supérieur à 8192,
- la somme n° de secteur et nombre de secteurs est supérieure à 8192.

```
%CPSMD    SNIV = ?    ADR = ?    ITN = ?    IOP = ?
```

Cette macro décrit une unité d'échange multiplexée disque à interface SMD laquelle peut comprendre jusqu'à quatre drives.

Signification des paramètres :

- **SNIV : est un nombre décimal représentant le sous-niveau d'interruption sur lequel est connectée l'unité d'échange multiplexée.**
- **ADR : est un nombre hexadécimal représentant l'adresse du coupleur (exemple : '38).**
- **ITN : est un nombre décimal représentant le numéro d'interruption canal.**
- **IOP : est un nombre décimal représentant le numéro du processeur d'entrée-sortie qui gère le canal. Il peut varier de 0 à 3, s'il est omis, c'est le processeur de calcul(0) qui est pris par défaut.**

Remarque :

La gestion d'espace étant obligatoire sur ce type de disque, les macro-instructions suivantes seront du type "FUI" et "FUESP" (cf. chapitre 4.3.7).

```
%CPSAS    SNIV = ?    MODE = ?    ADR = ?    ITN = ?    IOP = ?
```

Cette macro décrit une unité d'échange multiplexée disque "WINCHESTER".

Paramètres et remarque identiques à ceux décrits pour la macro %CPSMD ci-dessus.

Le paramètre MODE indique le type de canal (MDC ou HDC).

```
%PUFDD   SNIV = ?   MODE = ?   ADR = ?   ITN = ?   IOP = ?
```

Cette macro-instruction décrit une unité d'échange pouvant contenir 1 à 4 unités de disques souples.

SNIV : représente le sous-niveau sous lequel est connecté l'unité physique
MODE : représente le mode de fonctionnement du canal. Il ne peut être que HDC, MDC ou LDC
ADR : représente l'adresse coupleur
ITN : représente le niveau d'interruption canal
IOP : représente le numéro du processeur d'E/S.

```
%FUFDD  ?   VOIE = ?   FINDIC = ?   ADRCYL = ?   NBCYL = ?   FACE = ?
```

Cette macro-instruction définit l'unité fonctionnelle disque souple.

FUFDD : précise le numéro de FU (décimal ou symbolique)
VOIE : précise le numéro de voie et varie de 0 à 3
FINDIC : précise comment le driver traite les secteurs de 128 octets pour les formats SEMS.

FINDIC = 1 -> tous les secteurs sont de 256 octets donc lorsqu'une disquette est formée de secteurs de 128 octets (ex : face 0 piste 0 pour les disquettes double densité ou les disquettes simple densité avec des secteurs de 128 octets), pour l'utilisateur tout se passe comme si la disquette était formée de secteurs de 256 octets.
Pour l'utilisation de FMS ou FDM, FINDIC doit être égal à 1.

FINDIC = 0 -> les secteurs ont leur taille réelle.
ADRCYL: précise l'adresse début de l'unité fonctionnelle en cylindres et peut varier de 0 à 74.
NBCYL : précise le nombre de cylindres de l'unité fonctionnelle peut varier de 1 à 75. NBCYL + ADRCYL £ 75.
FACE : précise le type de drive utilisé.
FACE = 1 -> drive 1 face
FACE = 2 -> drive 2 faces.

4.3.5 - Les macro-instructions “standard” périphérique

Ces macro-instructions servent à définir des périphériques conventionnels dit “standard”.
Si l’utilisateur a dans son installation ce genre de périphériques, il peut éviter de les décrire à l’aide des macro-instructions décrites dans les paragraphes précédents en utilisant les macro-instructions suivantes :

```
%TTY [TER30] [NOTIMOUT]
```

Description d’une télétype ou d'une terminet "standard".

Signification des paramètres

Les paramètres TER30 et NOTIMOUT sont optionnels.

- Paramètre TER30 :

. présent : description d’une terminet "standard" et des FU 2 et 3
. absent : description d’une télétype "standard" et des FU 1 à 4.

- Paramètre NOTIMOUT :

. présent : il y a non surveillance du clavier généré
. absent : il y a surveillance du clavier généré.

Cette macro-instruction n’est autorisée que si la TTY est définie comme ci-après :

Macro	Niveau	S/Niveau	Mode	Adr.	ITN	IOP
%TTY---	15	0	PP	'17F8	/	/

Description des unités fonctionnelles définies par les macros

MACRO	Nom	Numéro	CDE	SENS	FINDIC
%TTY	TR	1	'41	I	1
	TS	2	'45	O	0
	TK	3	'61	I	0
	TP	4	'45	O	1
%TTY TER30	TS	2	'845	O	0
	TK	3	'861	I	0
%TTY TER30 NOTIMOUT	TS	2	'845	O	0
	TK	3	'C61	I	0
%TTY NOTIMOUT	TR	1	'41	I	1
	TS	2	'45	O	0
	TK	3	'461	I	0
	TP	4	'45	O	1

% H R

Description d'un lecteur rapide de bandes perforées et de la FU5.

Cette macro-instruction n'est autorisée que si le lecteur rapide est défini comme ci-après :

MACRO	Niveau	S/Niveau	Mode	Adr.	ITN	IOP
%HR	15	4	PP	'8	/	/

Description de l'unité fonctionnelle définie par %HR

MACRO	Nom	Numéro	CDE	SENS	FINDIC
%HR	HR	5	'1	I	0

%HP

Description d'un perforateur rapide et de la FU6.

Cette macro-instruction n'est autorisée que si le perforateur rapide est définie comme ci-après :

MACRO	Niveau	S/Niveau	Mode	Adr.	IITN	IOP
%HP	15	5	PP	'C	/	/

Description de l'unité fonctionnelle définie par %HP

MACRO	Nom	Numéro	CDE	SENS	FINDIC
%HP	HP	6	'1	O	0

%CR [IOP = ? [RACK = ?]

Description d'un lecteur de carte et de la FU7.

Signification des paramètres :

Les paramètres IOP = ? et RACK = ? sont optionnels.

- Paramètre "IOP" (IOP ?)

Ce paramètre précise le numéro de processeur qui gérera les entrées-sorties ; ce paramètre vaut 0, 1, 2. 3. S'il est omis, le processeur de calcul (CPU n° 0) se chargera de gérer les entrées-sorties.

- Paramètre "RACK" (RACK = ?)

Ce paramètre ne peut être décrit que si on a défini un numéro de IOP.
Ce paramètre précise si le coupleur se trouve dans le rack extension (OFF) ou pas (ON). Il est pris "ON" par défaut.

Syntaxe :

%CR [IOP = 9 [RACK = {ON / OFF}]]

Erreurs détectées :

L'erreur "paramètre Incorrect" est détectée si :

- pour le paramètre "IOP"
  . il est supérieur à 3,
- pour le paramètre "RACK"
  . il est différent de ON, OFF.

Cette macro-instruction n'est autorisée que si le lecteur de cartes est défini comme ci-après :

MACRO	Niveau	S/Niveau	Mode	Adr.	ITN	IOP
%CR	14	4	LDC	'10	4	0
%CR IOP = x	14	4	LDC	'10	4	X
%CR IOP = x RACK = OFF	14	4	LDC	'810	4	X

Description de l'unité fonctionnelle définie par la macro %CR (quelque soient les paramètres associés).

MACRO	Nom	Numéro	CDE	SENS	FINDIC
%CR - - -	CR	7	'0	I	0

```
%LP     [IOP = ? [RACK = ?]]
```

Description d'une imprimante rapide et de la FU8.

Les paramètres, la syntaxe, et les erreurs sont les mêmes que dans la macro %CR.

Cette macro-instruction n'est autorisée que si l'imprimante est définie comme ci-après :

MACRO	Niveau	S/Niveau	Mode	Adr.	ITN	IOP
%LP	14	7	LDC	'40	7	0
%LP IOP= x	14	7	LDC	'40	7	x
%LP IOP= x RACK = OFF	14	7	LDC	'840	7	x

Description de l'unité fonctionnelle définie par la macro %LP (quelque soient les paramètres associés).

MACRO	Nom	Numéro	CDE	SENS	FINDIC
%LP - - -	LP	8	'45	O	0

```
%MT BPI = ?   MODE = ?   [IOP= ? [RACK= ?]]   [NBV = ?]
```

Description d'un coupleur bande magnétique.

Les paramètres "IOP", "RACK", "NBV" sont optionnels.

Signification des paramètres :

- Paramètre "BPI" (BPI = ?)

Ce paramètre précise la densité employée : 1600 BPI ou 800 BPI ; il ne doit pas être omis.

- Paramètre "Mode" (MODE = ?)

Ce paramètre précise le mode de fonctionnement du périphérique. Il vaut :

. LDC (Low Speed Data Channel)
. MDC (Medium Speed Data Channel)
. HDC (High Speed Data Channel).

il est obligatoire.

- Paramètre "IOP" (IOP = ?)

Ce paramètre précise le numéro du processeur qui gérera les entrées-sorties.
Ce paramètre vaut 0, 1, 2, 3. S'il est omis, le processeur de calcul (CPU n° 0) se chargera de gérer les entrées-sorties.

- Paramètre "RACK" (RACK ?)

Ce paramètre ne peut être décrit que si on a défini un numéro de IOP.
Ce paramètre précise si le coupleur se trouve dans le rack extension (OFF) ou pas (ON). Il est pris "ON" par défaut.

- Paramètre "NBV" (NBV ?)

Ce paramètre précise le nombre de voies utilisées sur le coupleur que l'on définit. Il vaut 1 ou 2 et 2 par défaut. S'il vaut :

. 1 on génère une unité fonctionnelle FU9
. 2 on génère 2 unités fonctionnelles FU9 et FU10.

Syntaxe :

$$\%MT \quad BPI=\begin{Bmatrix}800\\1600\end{Bmatrix} \quad MODE=\begin{Bmatrix}LDC\\MDC\\HDC\end{Bmatrix} \quad \left[IOP=9 \quad \left[RACK=\begin{Bmatrix}ON\\OFF\end{Bmatrix}\right]\right] \left[NBV=\begin{Bmatrix}2\\1\end{Bmatrix}\right]$$

Erreurs détectées :

L'erreur "paramètre incorrect" sera détectée si :

- pour le paramètre "BPI"
  . il est différent de 800 ou 1600

- pour le paramètre "Mode"
  . il est différent de HDC, MDC, LDC

- pour le paramètre "IOP"
  . il est supérieur à 3

- pour le paramètre "RACK"
  . il est différent de ON ou OFF

- pour le paramètre "NBV"
  . il est différent de 1 ou 2.

Cette macro-instruction n'est autorisée que si le dérouleur de bande magnétique est définie comme ci-après :

MACRO	Niveau	S/Niveau	Mode	Adr.	ITN	IOP
%MT BPI = ? MODE = xDC	0 à 15	2	xDC	'18	2	0
%MT BPI = ? MODE = xDC IOP = x	0 à 15	2	xDC	'18	2	x
%MT BPI = ? MODE = xDC IOP = x RACK = OFF	0 à 15	2	xDC	'818	2	x

Description des unités fonctionnelles par la macro %MT

MACRO	Nom	Numéro	CDE	SENS	FINDIC
%MT BPI = 800 - - - NBV = 1	T1	9	'0	IO	1
%MT BPI = 1600 - - - NBV = 1	T1	9	'0	IO	0
%MT BPI = 800 - - - [NBV = 2]	T1 T2	9 10	'0 '0	IO IO	1 1
%MT BPI = 1600 - - - [NBV = 2]	T1 T2	9 10	'0 '0	IO IO	0 0

4.3.6 - Configuration horloge temps réel

```
%HTR      SNIV = ?      FREQ = ? [ADR = ? ]
```

Cette macroinstruction sert à définir l'horloge temps réel. Elle ne peut être utilisée qu'une seule fois.

Signification des paramètres

- SNIV indique le sous-niveau des interruptions horloge. C'est un nombre décimal inférieur à 16
- FREQ indique la fréquence des interruptions horloge (en nombre de tops par seconde). C'est un nombre décimal inférieur ou égal à 1000.
- ADR représente l'adresse de l'horloge (nombre hexadécimal). Si ce paramètre est omis, ADR = '17FC.

Syntaxe :

```
%HTR    SNIV = 99      FREQ = 999    [ ADR = 'FFFF]
```

Erreurs détectées :

- "paramètre incorrect" si :
  . SNIV supérieur à 15
  . FREQ supérieur à 1000.

- "macro interdite" si ce n'est pas la première macro %HTR rencontrée.

Remarques :

- la macro % HTR définit une unité fonctionnelle gérée de manière spécifique par IOCS.
- le driver correspondant sera en général intégré dans le superviseur.

4.3.7 - Les macro-instructions pour la gestion de volume

- Macro à utiliser dans la phase de dimensionnement

%FSMON GVOL	Intégrer toutes les fonctions spéciales moniteur de gestion volume
%FSMON DMONT NUM = 8	Intégrer la fonction de démontage de volume
%FSMON MONT NUM = 9	Intégrer la fonction de montage de volume
%FSMON READ NUM = 10	Intégrer la fonction de lecture de la structure

Description détaillée des macro-instructions pour la gestion de volume

- Phase de dimensionnement

%FSMON GVOL

Cette macro-instruction permet de définir les 3 fonctions spéciales moniteur non standard de gestion de volume.
Son écriture remplace l'écriture des 3 macro-instructions ci-après :

%FSMON DMONT NUM 8

Cette macro permet de définir la fonction spéciale moniteur non standard de démontage de volume.

%FSMON MONT NUM 9

Cette macro permet de définir la fonction spéciale moniteur non standard de montage de volume.

%FSMON READ NUM 10

Cette macro permet de définir la fonction spéciale moniteur non standard de lecture de la structure d'un support.

Erreurs détectées :

"Paramètre incorrect" est détectée dans des conditions décrites dans la macro-instruction %FSMON ? NUM ?

L'utilisation de ces macro-instructions nécessite l'emploi de la bibliothèque BIBVOL -:s.

- Phase configuration

```
%FUICD ?    VOIE = ?    FIXE = ?
```

Cette macro-instruction permet la description d'une unité fonctionnelle initiale disque 10 mb appartenant à l'un des disques de l'unité d'échange multiplexée définie précédemment par une macro-instruction PUCD ou PUVM. Elle génère une macro-instruction %FUCD.

N.B. : Un cylindre contient 48 secteurs, c'est-à-dire 6 Kmots.

Signification des paramètres :

Paramètres apparents :

- Paramètre "num FU"

Ce paramètre indique le n° de FU. Il peut être exprimé en décimal (< à 126) ou en symbolique.

- Paramètre "num Voie"

Ce paramètre précise le numéro de voie sur laquelle est connectée la FU.
Ce nombre peut prendre des valeurs de 0 à 3. Il doit être inférieur au nombre de voies défini dans la macro-instruction PUCD.

- Paramètre "Fixe"

Ce paramètre vaut Y si la FU se trouve sur le plateau fixe et N si elle se trouve sur le plateau mobile.

Paramètres implicites :

- Paramètre "Adresse cylindre"

Ce paramètre précise le numéro de cylindre où commence la FU disque.
Ce paramètre vaut 0.

- Paramètre "nombre cylindre"

Ce paramètre indique la longueur de la FU disque ou nombre de cylindres.
Ce paramètre vaut 400.

Syntaxe :

```
%FUICD { 999 } VOIE = 9 FIXE = { Y }
       { XX  }                 { N }
```

Erreurs détectées :

L'erreur "macro-instruction interdite" est détectée si cette macro-instruction n'a pas été précédée d'une macro-instruction PUCD ou PUVM.

L'erreur "paramètre incorrect" est détectée si :

- Pour le paramètre "num FU"

. de type décimal est supérieur à 125,
. de type symbolique non reconnu (voir § 2.6.1 du M.R. BOSD).

- Pour le paramètre "Voie"

. il n'est pas de type décimal
. il est égal ou supérieur au nombre de voies défini dans PUCD.

- Pour le paramètre "Fixe"

. il est différent de Y ou de N.

Remarques :

Deux unités fonctionnelles ne peuvent avoir le même numéro. Cette erreur n'est détectée qu'au moment de l'assemblage. Elle est fatale et nécessite de regénérer la configuration.

```
%FUIDP ? VOIE = ?
```

Cette macro décrit une unité fonctionnelle "disque de moyenne capacité" initiale appartenant à l'un des disques de l'unité d'échange multiplexée définie précédemment par une macro %PUDP. Elle génère une macro-instruction %FUDP.

N.B. : Un cylindre contient 480 secteurs c'est-à-dire 60 Kmots.
Les paramètres ont la même signification que ceux de la macro %FUICD.

```
%FUIVM ? VOIE = ?      FIXE = ?
```

Cette macro décrit une unité fonctionnelle "disque 20 méga-octets" initiale appartenant à l'un des disques de l'unité d'échange multiplexée définie précédemment par une macro %PUVM ou %PUCD. Elle génère une macro-instruction %FUVM.

NB : Un cylindre de la partie mobile contient 48 secteurs, c'est-à-dire 6 Kmots
Un cylindre de la partie fixe contient 144 secteurs, c'est-à-dire 18 K mots.

Les paramètres ont la même signification que ceux de la macro %FUICD.

```
%FUISMD ?    UNIT = ?    FIXE = ?
```

Cette macro-instruction décrit une unité fonctionnelle initiale d'un disque à interface SMD. Chacun des disques est identifié par son numéro d'unité. Ce dernier correspond au numéro affiché sur la face avant du drive.

Signification des paramètres :

- Paramètre "numfu" : numéro de la FU initiale. Il peut être exprimé en décimal (< 126) ou en symbolique (D1, D2, E1, etc. . .).

- Paramètre "UNIT" : Numéro d'unité décimal de 0 à 7.
Dans une configuration comportant plusieurs disques, les macros FUISMD doivent respecter l'ordre croissant des numéros d'unité.

- Paramètre "FIXE" : = Y pour la partie fixe d'un disque
= N pour la partie mobile.

Remarque :

- Doit être précédée d'une macro %CPSMD

- Ou d'une macro %FUESPSMD.

```
%FUISAS ? UNIT = ?
```

Cette macro-instruction décrit une unité fonctionnelle initiale d'un disque "WINCHESTER" associée à un numéro d'unité.

Signification des paramètres :

- Paramètre "numfu" : numéro de FU initiale. Il peut être exprimé en décimal (126) ou en symbolique (D1, D2. El, etc. . .).

- Paramètre "UNIT" : Numéro d'unité décimal (0 à 2).

%FUIFDD ? VOIE = ?

Cette macro-instruction décrit une unité fonctionnelle disque souple, double face double densité, initiale. Elle génère une macro-instruction %FUFDD.

NB : le cylindre 0 est formé de 39 secteurs de 128 mots (piste 0 face 0 13 secteurs de 128 mots, piste 0 face 1 26 secteurs de 128 mots).
Tous les autres cylindres font 52 secteurs.

%FUESPCD ?

%FUESPVM ?

Ces macro-instructions permettent la description d'une unité fonctionnelle disque 10 ou 20 mb appartenant à l'un des disques de l'unité d'échange multiplexée définie précédemment par une macro-instruction PUCD ou PUVM. Elle génère une macro-instruction %FUCD ou %FUVM.

Signification des paramètres :

- Paramètre "num Fu"

Ce paramètre indique le numéro de FU et peut être exprimé en décimal (< 126) ou en symbolique (D1, D2, etc. . .}.

Syntaxe :

%FUESP{CD / VM}{999 / XX}

Erreurs détectées :

L'erreur "macro-instruction interdite" est détectée si cette macro-instruction n'a pas été précédée d'une macro-instruction PUCD ou PUVM.

L'erreur "paramètre incorrect" est détectée si le paramètre "numfu" :

. de type décima! est supérieur à 125,
. de type symbolique non reconnu (voir § 2.6.1 du M.R. BOSD).

%FUESPFDD ?

Cette macro-instruction permet la description d'une unité fonctionnelle sur disquette double densité double face. Elle fait suite à une macro %FUIFDD. Elle génère une macro-instruction %FUFDD.
Le paramètre a la même signification que celui de la macro %FUESPCD.

%FUESPDP ?

Cette macro-instruction permet la description d'une unité fonctionnelle "disque de moyenne capacité" dynamique appartenant à l'un des disques de l'unité d'échange multiplexée définie précédemment par une macro %PUDP. Elle génère une macro-instruction %FUDP.

Le paramètre a la même signification que celui de la macro %FUESPCD.

%FUESPSAS ?

Cette macro-instruction permet la description d'une unité fonctionnelle disque "WINCHESTER" dynamique appartenant à l'un des disques de l'unité d'échange multiplexée définie précédemment par une macro %CPSAS.

Le paramètre a la même signification que celui de la macro %FUESPCD.

%FUESPSMD ?

Cette macro-instruction permet la description d'une unité fonctionnelle "disque à interface SMD" dynamique appartenant à l'un des disques de l'unité d'échange multiplexée définie précédemment par une macro %CPSMD.

Le paramètre a la même signification que celui de la macro %FUESPCD.

4.3.8 - Fin de génération

%ENDGEN

Cette macro-instruction termine la génération. Elle permet d'imprimer un EOT sur la bande objet.
Toute macro-instruction se trouvant donc après ENDGEN sera interdite.
Elle est obligatoirement suivie de la directive * END qui provoque le retour sous le contrôle du superviseur.

4.4 - EXEMPLE D'ECRITURE

4.4.1 - Configuration standard

1 lecteur de carte
1 imprimante
1 télétype
1 perforateur ruban
1 lecteur ruban

```
% OPTION = MULTI
% NIVEAU  14  KSTOR =
% CR
% LP
% NIVEAU  15  KSTOR =
% TTY
% HR
% HP
% ENDGEN
* END
```

4.4.2 - Configuration sans téléimprimeur

avec : — 1 console de visualisation sur coupleur MUX4
— lecteur et perforateur de ruban.

```
% OPTION = MONO
% NIVEAU  8  KSTOR =
% CPMUX4  MODE = PP  ADR = '17F8  IOP =  CONNEX= MUX4
% PUMUX4  VT  SNIV1 = 1  SNIV2 = 5  ITN =  VOIE = 3
% FU  2  CDE = '1  SENS = 0  FINDIC = 0
% FU  3  CDE = '1  SENS = 1  FINDIC = 1
% NIVEAU  15  KSTOR =
% HR
% HP
% ENDGEN
* END
```

4.4.3 - Configuration avec disque

Cette configuration est destinée à fonctionner sous BOS-D sans gestion de volume.
Elle comporte un disque à cartouche de 10 Mega octets, découpé en 3 unités fonctionnelles.

```
% OPTION = MULTI
% NTACHES = 48
% SUMAX = 20
% POOL  NBUF = 2  LBUF = 80
% NIVEAU  14      KSTOR =
% CR
```

```
%PUCD SNIV1 = 1 MODE = HDC ADR = '30 ITN = 1 IOP = NBV = 1
%FUCD D3 VOIE = 0 ADRCYL = 1 NBCYL = 399 FIXE = N
%FUCD D1 VOIE = 0 ADRCYL = 0 NBCYL = 50 FIXE = Y
%FUCD D2 VOIE = 0 ADRCYL = 50 NBCYL = 350 FIXE = Y
%LP
%NIVEAU 15 KSTOR =
%TTY
%NIVEAU 13 KSTOR =
%HTR SNIV = 0 FREQ = 50
%ENDGEN
*END
```

4.4.4 - Description d'un disque à têtes fixes

```
%PU DK SNIV = 0 MODE = HDC ADR = '38 ITN = 0 CONNEX = IOP = 1
%FUDK D3 ADRDK = 0 NSECT = 132
%FUDK D4 ADRDK = 192 NSECT = 8000
```

4.4.5 - Description d'un coupleur MXP04

Supposons une table traçante, et deux consoles de visualisation connectées sur un coupleur MXP04.

Les macros correspondantes seront :

```
%CPMUX4 MODE = PP ADR = '1300 IOP =
%PUMUX4 TAB SNIV1 = 4 SNIV2 = ITN = VOIE = 0
%FU CR CDE = '45 SENS = O FINDIC = 0
%PUMUX4 VT SNIV1 = 6 SNIV2 = 10 ITN = VOIE = 1
%FU F1 CDE = '1 SENS= IO FINDIC = 1
%PUMUX4 VT SNIV1 = 7 SNIV2 = 11 ITN = VOIE = 2
%FU F2 CDE = '1 SENS = IO FINDIC = 1
```

Remarque : le type de périphérique TAB doit avoir été décrit auparavant par une macro %DEFPU.

4.4.6 - Description d'un coupleur MXP16

Soit un coupleur MXP16 sur lequel sont connectées 10 téléimprimeurs ou consoles de visualisation.

```
%CPMPX P16 SNIV = 5 MODE = PP ADR = '1600 ITN = CONNEX = MUX16 IOP =
NBV = 16

%PUMPX VT VOIE = 0
%FU F1 CDE = '381 SENS = IO FINDIC = 1
%PUMPX VT VOIE =1

%PUMPX VT VOIE = 9
%FU FA CDE = '381 SENS = IO FINDIC = 1
```

Remarques :
Le paramètre NBV = 16 est obligatoire même si le nombre de voies utilisées est inférieur.
Le paramètre CDE = '381 indiqué dans l'exemple signifie :
8 bits/caractère
2 bits/stop
Validation des interruptions.

4.4.7 - Description d'un coupleur MXP8

Soit un coupleur MXP8 sur lequel sont connectées 5 consoles de visualisation.

```
% CPMPX  P16  SNIV=5  MODE=PP  ADR='1600  ITN=  CONNEX=MUX8  IOP=
NBV=8
% PUMPX  VT  VOIE=0
% FU  F1  CDE='381  SENS=IO  FINDIC=1
% PUMPX  VT  VOIE=1
          -
          -
          -
          -
% PUMPX  VT  VOIE=4
% FU  F5  CDE='381  SENS=IO  FINDIC=1
```

Remarque : le mot de commande a la même signification que pour le MXP16.
Le paramètre NBV=8 est obligatoire même si le nombre de voie utilisées est inférieur.

4.4.8 - Description d'un coupleur disque souple (FTB)

Soit une configuration comprenant deux unités de disque souple.

```
% DEFPU  FD0  CODE=16  LZONE=30  SP=
          -
          -
          -
% NIVEAU  14  KSTOR=
% PU  FD0  SNIV=0  MODE=LDC  ADR='28  ITN=0  CONNEX=  IOP=
% FU  D1  CDE='0  SENS=IO  FINDIC=0
% FU  D2  CDE='100  SENS=IO  FINDIC=0
```

Remarque : L'octet gauche du mot de commande indique le numéro d'unité.

4.4.9 - Description d'un coupleur disque souple (FIB)

Soit une configuration comprenant deux unités de disque souple.

```
% DEFPU  FD1  CODE=16  LZONE=11  SP=
          -
          -
          -
% NIVEAU  14  KSTOR=
% PU  FD1  SNIV=0  ADR='28  ITN=0  CONNEX=  IOP=
% FU  D1  CDE='0  SENS=IO  FINDIC=1
% FU  D2  CDE='100  SENS=IO  FINDIC=1
```

Remarque : L'octet gauche du mot de commande contient le numéro d'unité.
Si FINDIC = 1 les secteurs adressés sont des secteurs de 128 mots.

4.4.10 - Description d'un coupleur bande magnétique

Soit une configuration comprenant deux dérouleurs de bande magnétique.

a) en utilisant la macro %CPMPX

```
% CPMPX  MT  SNIV=2  MODE=HDC  ADR='18  ITN=2  CONNEX=  IOP=  NBV=2
% PUMPX  MT  VOIE=0
% FU  T1 CDE='0  SENS=IO  FINDIC=1
% PUMPX  MT  VOIE=1
% FU  T2  CDE='0  SENS= IO  FINDIC=1
```

Remarque : FINDIC = 0 pour une densité de 1600 BPI.
= 1 pour une densité de 800 BPI.

b) en utilisant la macro simplifiée %MT

```
% MT BPI = 800 MODE = HDC
```

4.4.11 - Intégration des Fonctions Spéciales gérant le montage de volumes : génération statique

```
%FSMON  MONT  NUM= 9
%FSMON  READ  NUM = 10
%FSMON  DMONT  NUM = 8
%NIVEAU  13  KSTOR=
%HTR  SNIV = 0  FREQ= 50
%NIVEAU  14  KSTOR =
%LP
%PUCD  SNIV = 1  MODE = HDC  ADR = '30  ITN = 1  IOP=  NBV = 1
%FUCD  D1  VOIE = 0  ADRCYL = 0  NBCYL = 40  FIXE = Y
%FUCD  D2  VOIE = 0  ADRCYL = 40  NBCYL = 242  FIXE= Y
%FUCD  D8  VOIE = 0  ADRCYL = 288  NBCYL = 112  FIXE= Y
%FUCD  E5  VOIE = 0  ADRCYL = 0  NBCYL= 400  FIXE= N
%FUCD  D3  VOIE = 0  ADRCYL = 1  NBCYL = 399  FIXE= N
%FUCD  DA  VOIE = 0  ADRCYL = 40  NBCYL = 242  FIXE= N
%FUCD  D9  VOIE = 0  ADRCYL = 100  NBCYL= 300  FIXE = N
%FUCD  D4  VOIE = 0  ADRCYL = 282  NBCYL = 6  FIXE= Y
%FUCD  D5  VOIE = 0  ADRCYL = 1  NBCYL = 300  FIXE= N
%FUCD  E4  VOIE = 0  ADRCYL = 200  NBCYL = 200  FIXE = N
%FUCD  D6  VOIE = 0  ADRCYL = 301  NBCYL = 99  FIXE = N
%FUCD  D7  VOIE = 0  ADRCYL = 0  NBCYL = 400  FIXE= Y
%FUCD  E3  VOIE = 0  ADRCYL = 0  NBCYL = 200  FIXE = N
%FUCD  E1  VOIE = 0  ADRCYL = 1  NBCYL = 3  FIXE = N
%FUCD  E2  VOIE = 0  ADRCYL = 4  NBCYL = 196  FIXE = N
```

Commentaires : L'unité fonctionnelle "E5" est la FU initiale du plateau mobile. Sa présence est impérative et elle doit être la première décrite. Elle débute sur le cylindre 0 et sa longueur est égale à la taille du support.

4.4.12 - Description de systèmes BOSD utilisant la gestion de volume

Cet exemple décrit une gestion d'espace dynamique.

La configuration comprend :

- un disque à cartouche support système,
- un dérouleur de bande magnétique 800 BPI,
- une imprimante,
- une télétype TER30,
- un lecteur de carte.

Dans cet exemple on utilise les macro-instructions standard système et périphérique afin de simplifier l'écriture des macro-instructions.

```
<-------------------------------------
<                    FICIOC-BG
<-------------------------------------
%BOSD
%POOL  NBUF = 4  LBUF =80
<---------------- MONTAGE VOLUME ----------
%FSMON  GVOL
<
<---------------- NIVEAU  14 ----------------
%NIVEAU  14  KSTOR = 30
<---------------- DISQUE  VM ----------------
%PUCD  SNIV = 1  MODE = HDC  ADR = '30  ITN = 1  IOP = 1  NBV = 1
%FUICD DB  VOIE = 0  FIXE = N
%FUCD  D1  VOIE = 0  ADRCYL = 000  NBCYL = 400  FIXE = N
%FUCD  D2  VOIE = 0  ADRCYL = 004  NBCYL = 196  FIXE = N
%FUICD DA  VOIE = 0  FIXE = Y
%FUESPCD  D3
%FUESPCD  D4
%FUESPCD  D5
%FUESPCD  D6
%FUESPCD  D7
%FUESPCD  D8
%FUESPCD  D9
%FUCD  DF  VOIE = 0  ADRCYL = 000  NBCYL = 001  FIXE= N
%FUCD  E1  VOIE = 0  ADRCYL = 001  NBCYL = 003  FIXE= N
%FUCD  E3  VOIE = 0  ADRCYL = 000  NBCYL = 200  FIXE= N
%FUCD  E4  VOIE = 0  ADRCYL = 200  NBCYL = 200  FIXE= N
<
<---------------- BANDE MAGNETIQUE ----------
%MT  BPI = 800  MODE = MDC
<---------------- LECT. CARTE, IMPRIMANTE -------
%CR
%LP
<
<---------------- NIVEAU 15 -----------------
%NIVEAU  15
%HR
%HP
%TTY  NOTIMOUT
%ENDGEN
```

Ce deuxième exemple décrit une gestion d'espace dynamique.

La configuration comprend :

- un disque à cartouche support système,
- une unité disque 50 méga-byte,
- une imprimante,
- un dérouleur BM 800 BPI,
- une télétype TER30.

Dans cet exemple on utilise les macro-instructions standard système et périphérique afin de simplifier l'écriture des macro-instructions.

```
%BOSD
%FSMON   GVOL
%POOL NBUF = 4 LBU F = 80
<
%NIVEAU 13
%HTR SNIV = 0 FREQ = 50
<
%NIVEAU 14
<
<       DISQUE 400
<
%PUCD SNIV = 1 MODE = HDC ADR = 430 ITN = 1 IOP = NBV = 1
%FUICD D7 VOIE = 0 FIXE= Y
%FUCD D1 VOIE = 0 ADRCYL = 0 NBCYL = 40 FIXE = Y
%FUCD D2 VOIE = 0 ADRCYL = 40 NBCYL = 242 FIXE = Y
%FUCD D4 VOIE = 0 ADRCYL = 282 NBCYL = 118 FIXE = Y
%FUICD D9 VOIE = 0 FIXE= N
%FUESPCD D3
%FUESPCD D5
%FUESPCD D6
<
<             DISPACK
<
%PUDP SNIV = 0 MODE = MDC ADR = '38 ITN = 0 IOP = NBV = 1
%FUIDP E6 VOIE= 0
%FUESPDP F1
%FUESPDP F2
%FUESPDP F3
%FUESPDP E1
%FUESPDP E2
%LP
%CR
<       BANDE T1 T2
<
%MT BPI = 800 MODE = MDC
<
%NIVEAU 15
%TTY TER30 NOTIMOUT
<
%ENDGEN
*END
```

Remarque :

Le plateau fixe du disque à cartouche est généré en statique car il supporte le disque système (D1, D2) et ne subit pas la reconfiguration dynamique. Cependant, la FU D7 décrit la FU initiale du support pour permettre de le structurer et de travailler avec les FUP.

5 - MISE EN OEUVRE

REMARQUE:

Pour être utilisable sur de petites configurations ne disposant pas de disque, la bibliothèque de macro-définitions GENIO est découpée en 3 parties.

On pourra n'utiliser que la première ou les deux premières parties suivant la taille mémoire disponible et les macroinstructions utilisées.

- la première partie contient les macro-définitions suivantes :

%ØPTIØN
%NTACHES, %SUMAX, %INPMAX
%PØØL, %FSMØN, %DEFPU, %SYMBEXT
%NIVEAU
%PU, %FU, %TUP
%CPMUX4, %PUMUX4
%ENDGEN

- la deuxième partie contient :
%CPMPX, %PUMPX, %PUDX
%TTY, %HR, %HP, %CR, %LP
%HTR

- la troisième partie contient :
%PUCD, %FUCD, %FUDK, %PUDP, %FUDP, %PUVM, %FUVM, %PUIB, %FUIB, %PUFDD, %FUFDD.

- la lecture de la première partie se fera par une commande IMAC adressée à MACP

- la deuxième partie sera lue à la suite d'une commande CMAC

- la troisième partie sera lue après une autre commande CMAC.

En utilisant BOS-A et la première partie de GENIO, il est possible de générer IOCS sur une configuration comportant 8 K mots de mémoire.

5.1 - MISE EN OEUVRE SOUS BOS A/B

- chargement de MACP au moyen du chargeur translateur.
- affectation des unités symboliques

  *LØ ZE
  *SI HR pour lecture de GENIØ sur le lecteur rapide de ruban.
  *SØ HP pour obtenir le texte objet de MACP sur le perforateur

- lecture de la première partie

  *IMAC

- éventuellement lecture de la deuxième partie de GENIO

  *CMAC

- affectation de l'unité symbolique de lecture des macroinstructions.

  *SI { TK | HR | CR }

- lecture des macroinstructions

  *CMAC

- chargement de l'assembleur par le chargeur translateur (ASM ou ASM-E)
- assemblage-des tables de IOCS produites par GENIO

  *LØ ZE
  *SI HR
  *BØ HP
  *IASM

- assemblage du noyau de IOCS (mono ou multi-tâche)

  *SI HR
  *CASM

- le ruban objet produit par l'assembleur contient le noyau de IOCS et ses tables. Il est à relier par édition de liens à un superviseur et aux drivers nécessaires.
  Les drivers nécessaires sont ceux dont les noms sont donnés dans les macroinstructions.

  %PU
  %CPMPX
  %PUMPX
  %PUMUX4
  %PUDX

5.2 - MISE EN OEUVRE SOUS BOS-D

*CALL MACP
* LØ ZE
* SI { fichier | HR | ⋮ } support de GENIO

```
*IMAC
*CMAC                          Lecture des 3 parties de GENIØ
CMAC
*SI      fichier               support des macroinstructions
         HR
         CR
*CMAC
*CALL ASM
*BØ      fichier support de IOCS à link-éditer
*IASM
*SI      fichier support du noyau de IOCS (mono ou multi-tâche)
*CASM
*EØJ
```

5.3 - LISTE DES MACROINSTRUCTIONS

```
% ØPTIØN = {MØNØ | MULTI}
% NTACHES = 999
%SUMAX = 999
%INPMAX = 9999
%PØØL NBUF = 99 LBUF = 999
%FSMØN XXXXX NUM = 999
%DEFPU XXX CØDE = {8 | 16} LZØNE = 999 SP = [{Y | N}]
%SYMBEXT xxxxxx
%BOSD
%RTESM
%RTESD
%MPES
%TSM
%MUTEX
%NIVEAU 99 KSTØR = [999]
%PU XXX SNIV = 99 MØDE = {PP | LDC | MDC | HDC} ADR = 'FFFF ITN = [99]
                         CONNEX = [{STD | ASYV1}] IOP = 9
%TUP-99 = xxxxxx
% FU{999 | XX} CDE = 'FFFF SENS = {I | Ø | IØ} FINDIC = {0 | 1}
%CPMPX XXX SNIV = 99 MØDE = {PP | LDC | MDC | HDC} ADR = 'FFFF ITN = [99]
                           CONNEX = [{MUX8 | MUX16}] IOP = 9 NBV = 9
%PUMPX XXX VØIE = 99
%PUDX xxx BEGIN = 99 NUMER = 99 STEP = 9 ADR = 'FFFF
```

% CPMUX4 MØDE = {PP, LDC, MDC, HDC} ADR = 'FFFF IØP = [9] [CONNEX = [{STD, MUX4, ASIV1}]]

% PUMUX4 XXX SNIV1 = 99 SNIV2 = [99] ITN = [99] VØIE = 9

% PUCD SNIV = 99 MØDE = {MDC, HDC} ADR = 'FFFF ITN = 99 IØP = [9] NBV=9

% FUCD {999, XX} VØIE =9 ADRCYL = 999 NBCYL = 999 FIXE = {Y, N}

% PUVM SNIV = 99 MODE = {MDC, HDC} ADR = 'FFFF ITN = 99 IOP = [9] NBV = 9

% FUVM {999, XX} VOIE = 9 ADRCYL = 999 NBCYL = 999 FIXE = {Y, N}

% PUDP SNIV = 99 MODE = {MDC, HDC} ADR = 'FFFF ITN = 99 IOP = [9] NBV = 9

% FUDP {999, XX} VOIE = 9 ADRCYL = 999 NBCYL = 999

% FUDK {999, XX} ADRDK = 9999 NSECT = 9999

% TTY [TER30] [NOTIMOUT]
% HR
% HP
% CR [IOP = 9 [[RACK = {ON, OFF}]]]
% LP [IOP = 9 [[RACK = {ON, OFF}]]]
% MT BPI
% MT BPI = {800, 1600} MODE = {LDC, MDC, HDC} IOP = 9 [RACK = {OFF, ON}] [NBV = {1, 2}]

% FSMON GVOL

% FUICD {999, XX} VOIE = 9 FIXE = {Y, N}

% FUESPCD {999, XX}

% FUIVM {999, XX} VOIE = 9 FIXE = {Y, N}
% FUESPVM {999, XX}
% FUIDP {999, XX} VOIE = 9
% FUESPDP {999, XX}
% HTR SNIV = 99 FREQ = 999

% PUIB SNIV = 99 MODE = {LDC, MDC, HDC} ADR = 'FFFF ITN = 99 IOP = [9] NBV = 9

% FUIB {999, XX} VOIE = 9 SENS = {I, O, IO} FINDIC = 1 ADRSEC = 9999 NBSEC = 9999

```
% PUFDD  SNIV= 99  MODE=(LDC|MDC|HDC)      ADR= `FFFF  ITN= 99  IOP= 9

% FUFDD (999|XX)  VOIE= 9  FINDIC= 9  ADRCYL= 99  NBCYL= 99  FACE= 9
% FUIFDD (999|XX) VOIE= 9
% FUESPFDD (999|XX)
% CPSMD  SNIV= 99  ADR= `FFFF  ITN= 99  IOP  9
% FUISMD (999|XX)  UNIT= 99  FIXE=(Y|N)
% FUESPSMD (999|XX)
% CPSAS   SNIV= 99   MODE= (MDC|HDC)      ADR= `FFFF   ITN= 99   IOP= 9
% FUISAS  (999|XX)      UNIT= 9
% FUESPSAS    (999|XX)

% ENDGEN
```

5.4 - PARAMETRES DES MACROINSTRUCTIONS

Peripheriques	Abré-viation	Nbre de FU Mini	Nom FU associé	CDE	LZONE	FINDIC	CODE
Téléimprimeur	TTY	2/4	TR TS TK TP	'0041 '0045 '0061 '0045	0 0 0 0	1 0 0 1	8 8 8 8
Lecteur de ruban	HR	1	HR	'0025	0	0	8
Perforateur de ruban	HP	1	HP	'0001	0	0	8
Imprimante	LP	1	LP	'0045	0	0	8
Disque 10 mb	CD	1		'0000	26	0	16
Disque 20 mb	VM						
Disque à têtes fixes	DK	1		'0000	0	0	16
Disque SMD	SMD	1		'0000	90	0	16
Lecteur de cartes	CR	1	CR	'0000	0	0	16
Disque Winchester	SAS	1		'0000	68	0	16
Téléimpr./Visualisat. par MXP04	VT	1 par voie		'0001	3	(2)	8
Téléimpr./Visualisat. par MXP08 ou MXP16	VT	1 par voie		'0001 + (4)	3	0	8
Téléimpr./Visualisat. par Asynchrone 1 voie	VT	1		'0001 + (4)	3	(2)	8
Disque souple TE	FDO	1		(3)	30	0	16
Disque souple IBM	FD1	1		(3)	11	(6)	
Bandes magnéti-ques	MT	1 par voie	T1àT4	'0	3	(5)	16
Disque moy. capac.	DP	7		'0000	15	0	16
Floppy IBM multi-FU	IB	1		'0000	21	(7)	16
Floppy multifaces	FDD	1		'0000	52	(6)	16

(2) 0 si non échoplex 1 si échoplex pour une entrée
(3) Octet gauche = numéro d'unité (0 à 3)
(4) Format du caractère
(5) 3 = 0 = 1000 BPI 1 = 800 BPI
(6) 0 = secteurs de 64 Mots 1 = secteurs de 128 Mots (utilisation standard)
(7) = 1 : taille secteur = 128 mots.

5.5 - CONFIGURATION DES PERIPHERIQUES DEFINIS PAR LES MACRO STANDARD

On peut utiliser les macro-instructions standard si les périphériques sont configurés comme indiqué dans le tableau.

MACRO	Niveau	S/Niveau	Mode	Adr.	ITN	IOP
%TTY - - -	15	0	PP	'17F8	/	/
%HR	15	4	PP	'8	/	/
%HP	15	5	PP	'C	/	/
%CR	14	4	LDC	'10	4	0
%CR IOP = x	14	4	LDC	'10	4	X
%CR IOP = x RACK = OFF	14	4	LDC	'810	4	X
%LP	14	7	LDC	'40	7	0
%LP IOP= x	14	7	LDC	'40	7	X
%LP IOP= x RACK = OFF	14	7	LDC	'840	7	X
%MT BPI = ? MODE = xDC	0 à 15	2	xDC	'18	2	0
%MT BPI = ? MODE = xDC IOP = x	0 à 15	2	xDC	'18	2	X
%MT BPI = ? MODE = xDC IOP = x RACK = OFF	0 à 15	2	xDC	'818	2	X

Réseaux
et systèmes
d'information

Bull

Distribution codes/Codes de diffusion			
Customers : **Clients :**			
Internal : **Interne :**			

DELIVERY ADDRESS

ÉTIQUETTE ADRESSE

Bull Sems

1, Rue de Provence
B.P. 208
38432 ÉCHIROLLES CEDEX / FRANCE

Bull

Sems