

SOLAR

IOCS1

Moniteur de gestion des entrées-sorties

LOGICIEL

LOGICIEL

LOGICIEL

LOGICIEL

LOGICIEL

SOLAR

ENTREES - SORTIES

IOCS16

Logiciel

SUJET : Moniteur d'Entrées/Sorties des SOLAR 16-35.
16-70, 16-90

OBSERVATION :

VERSION LOGICIEL :

DATE : MAI 1985

(C) Bull-Sems 1985

Imprimé en France

Vos suggestions sur la forme et le fond de ce manuel seront les bienvenues. Une feuille destinée à recevoir vos remarques se trouve A la fin du présent manuel.

Ce document est fourni à titre d'information seulement. Il n'engage pas la responsabilité de Bull-Sems en cas de dommages résultant de son application. Des corrections ou modifications au contenu de ce document peuvent intervenir sans préavis; des mises à jour ultérieures les signaleront éventuellement aux destinataires.



1	PRESENTATION	1.1
2	DESCRIPTION GENERALE D'IOCS16	2.1
	2.1 INTEGRATION A UN SYSTEME	2.1
	2.2 INDEPENDANCE DES PROGRAMMES ET DES OPERATIONS D'ENTREE-SORTIES	2.1
	2.3 CARACTERISTIQUES GENERALES	2.2
3	LES FONCTIONS D'IOCS16	3.1
	3.1 DEMANDE D'ECHANGE EFFECTIF	3.2
	3.1.1 Forme de l'IOCB	3.2
	3.1.2 Description de l'IOCB	3.4
	3.2 FONCTIONS SPECIALES	3.10
	3.2.1 Forme de l'IOCB	3.10
	3.2.2 Description de l'IOCB	3.10
4	ECRITURE SYMBOLIQUE DE L'IOCB	4.1
	4.1 DEMANDE D'ECHANGE EFFECTIF	4.1
	4.1.1 Mot de fonction	4.1
	4.1.2 Type de retour à l'utilisateur	4.1
	4.2 FONCTIONS SPECIALES ADRESSEES AU MONITEUR	4.2
	4.3 FONCTIONS SPECIALES DE POSITIONNEMENT	4.2
5	LA FONCTION ATTENTE DE FIN D'OPERATION, D'ENTREE-SORTIE : WEIO	5.1
6	ACTIVATION DE LA TACHE ALARME	6.1
7	GESTION DU MONTAGE DE VOLUME	7.1
	7.1 GENERALITES	7.1
	7.2 FONCTIONNEMENT	7.2
	7.2.1 Définitions	7.2
	7.2.2 Principes de reconfiguration dynamique des tables de gestion du disque	7.2
	7.2.3 Gestion d'espace standard	7.4
	7.2.4 Gestion statique des disques	7.5
	7.2.5 Gestion d'espace préinitialisée	7.6
	7.3 DESCRIPTION	7.7
	7.3.1 Requête de démontage	7.7
	7.3.2 Requête de montage	7.8
	7.3.3 Requête d'identification d'un support :	7.10
8	DISPOSITIF "CHIEN DE GARDE IOCS16"	8.1



8.1 GENERALITES :	8.1
8.2 PRINCIPE DE FONCTIONNEMENT	8.1
8.2.1 Déclaration des FU surveillées :	8.1
8.2.2 Initialisation de la surveillance :	8.1
8.2.3 Déroulement d'un échange :	8.2

1 PRESENTATION

IOCS16 (Input Output Control System) est le moniteur d'entrées-sorties des unités centrales 16-35, 16-70 et 16-90 de la gamme SOLAR16.

Il assure le déroulement et le contrôle des entrées-sorties sur périphériques conventionnels (lecteur de cartes, de ruban, télétype etc...) et sur périphériques industriels (chaînes de mesures, entrées-sorties tout ou rien etc...).

Sa conception modulaire lui permet de s'adapter facilement à une installation par :

- l'intégration de modules gérant les échanges spécifiques sur les périphériques de l'installation et ne sélectionnant que ceux-ci.
- l'intégration de fonctions spéciales, au niveau du moniteur, propres à l'application.

IOCS16 est destiné aux calculateurs comportant l'option "scheduler microprogrammé MTS16".

Ses principales caractéristiques sont :

- la gestion des canaux LDC, MDC, HDC
- la gestion du DRPS
- l'horloge temps réel
- la gestion des IOP standards ou rapides (3 IOP pouvant être implantés dans le bac 0)
- le type de retour EMOD, IMOD, IBMOD)
- configurable par GI016.

Gestion des demandes :

Les demandes d'échange sont mises en file d'attente au niveau des unités physiques suivant la priorité des- tâches appelantes.

- chaque unité physique est considérée comme une ressource non partageable gérée par un sémaphore d'exclusion à un accès.
- toute tâche en attente de ressource est suspendue jusqu'à la libération de cette ressource.
- la synchronisation des tâches sur les fins d'échange est gérée par un sémaphore privé non paramétré réalisé sur le mot 4 de l'IOCB.

2 DESCRIPTION GENERALE D'IOCS16

2.1 INTEGRATION A UN SYSTEME

IOCS16 utilisé avec les systèmes d'exploitation SOLAR16 peut constituer l'un des éléments d'un système d'exploitation réalisé pour une application particulière.

2.2 INDEPENDANCE DES PROGRAMMES ET DES OPERATIONS D'ENTREE-SORTIES

L'interface unique entre le système d'entrées-sorties et l'utilisateur quel que soit le système d'exploitation permet une standardisation des séquences d'appel pour une opération d'E/S.

De plus la forme des requêtes est indépendante des configurations périphériques sur lesquelles les opérations sont effectivement réalisées.

IOCS16 reçoit des demandes portant sur des unités symboliques (SU) ou des unités fonctionnelles (FU) et adressées à des unités physiques (PU).

Unité Physique : une unité physique désigne un périphérique donné

Exemple : Télétype
Dériveur de bande magnétique

Unité Fonctionnelle : une unité fonctionnelle désigne un sous-ensemble d'un périphérique donné

Exemple Clavier télétype
Lecteur télétype
Portion de disque

Les liaisons unités fonctionnelles/unité physiques sont définies à la génération de l'IOCS16 concernant cette installation.

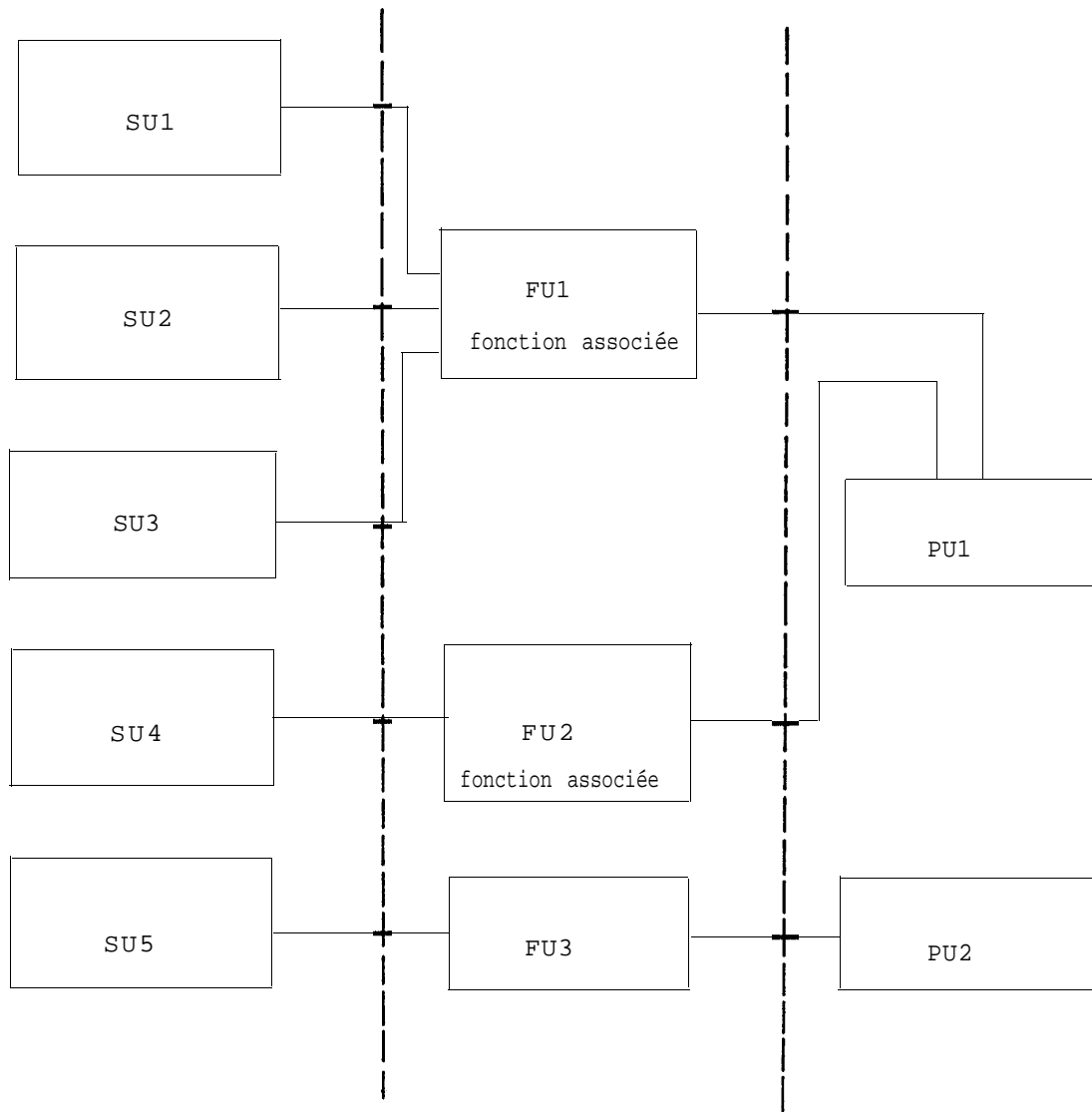
A chaque unité fonctionnelle est de plus associé un mot de commande qui sera envoyé au périphérique avant chaque échange de données. Plusieurs unités fonctionnelles peuvent être associées à une même unité physique.

Unité Symbolique : une unité symbolique désigne une unité d'échange caractéristique des programmes et non pas de la configuration de périphériques sur laquelle les opérations d'entrées-sorties sont effectivement réalisées.

Exemple Commande au superviseur
Entrée symbolique

Aux unités symboliques on associe par programme des unités fonctionnelles.

Ces liaisons sont dynamiques, elles peuvent être modifiées on-line (notamment par les dialogues des systèmes d'exploitation) sans avoir recours à une phase de définition de configuration système. Ces unités symboliques assurent pour de nombreux programmes une indépendance complète entre ces programmes et les opérations d'E/S qu'ils doivent réaliser.



2.3 CARACTERISTIQUES GENERALES

- Le programme IOCS16 est réentrant.
- Il se déroule sous le niveau de priorité de la tâche appelante. Celle-ci devra obligatoirement être une tâche software.
- L'appel à IOCS16 produit la destruction du registre X.
- IOCS16 utilise comme zone de travail la zone pointée par le registre K de l'appelant.
- L'utilisateur devra prévoir pour IOCS16 une zone de travail d'au moins 20 mots.

3 LES FONCTIONS D'IOCS16

On distingue deux types principaux de fonctions :

- les fonctions d'échange effectif, c'est-à-dire de transfert de données entre la mémoire et un périphérique
- les fonctions spéciales qui ne s'accompagnent pas d'un transfert de données. Celles-ci peuvent être subdivisées en deux catégories :
 - . Les fonctions d'intérêt général exécutées par le moniteur (FONCTION MONITEUR)
 - . Les fonctions de positionnement particulières à un périphérique exécutées par le "driver" associé à ce périphérique (FONCTION DE POSITIONNEMENT).

Séquence d'appel en assembleur :

Quelle que soit la fonction demandée la séquence d'appel est la suivante :

LAD	IOCB
SVC	IOCS

IOCB est la table des paramètres

IOCS est le numéro du sous-programme superviseur.

3.1.1 Forme de L'IOCB

Mot	-2	Extension		Optionnel
	-1	Extension		
A	→ 0	Octet de fonction	N° unité d'échange	
	1	Adresse de la table à transférer		
	2	Compte d'octets à transmettre ou adresse de la table des codes d'arrêt		
	3	Compte-rendu d'échange		Optionnel
	4	Evénement fin d'échange		
	5	Information complémentaire		

A) Echange avec buffer situé dans la zone de l'appelant. C'est l'appel réalisé- par un programme quel que soit son mode pour son propre compte.

A	→ 0	Octet de fonction	N° unité d'échange	
	1	Adresse de la table à transférer		
	2	Compte d'octets à transmettre ou adresse de la table des codes d'arrêt		
	3	Compte-rendu d'échange		Optionnel
	4	Evénement fin d'échange		
	5	Information complémentaire		

Les adresses de L'IOCB, de la table à transférer et de la table des codes d'arrêt sont des adresses absolues (mode maître) ou des adresses relatives (modes esclave et privilégié]. Dans ce cas elles sont situées dans la zone de l'appelant.

B) Echange avec buffer en CDA ou ZDR

IOCS16 offre la possibilité de gérer des buffers dans la CDA (COMMON DATA AREA) si cette option est disponible ou dans la ZDR (Zone des Données Résidentes). Dans ce cas les mémoires débanalisées d'adresse '18 et '19 contiennent les adresses début et fin de la CDA ou de la ZDR.

Mot -2	'FFFF		
-1	Octet de Fonction	N° unité d'échange	
A → 0	'FFFF		
1	Adresse de la table à transférer		
2	Compte d'octets à transmettre ou adresse de la table des codes d'arrêt		
3	Compte-rendu d'échange		
4	Evénement fin d'échange		
5	Information complémentaire		Optionnel

Les adresses de l'IOCB et de la table des codes d'arrêt sont des adresses absolues (appelant en mode maître) ou des adresses relatives (appelant en mode esclave ou privilégié).

Dans ce cas elles sont situées dans la zone de l'appelant. L'adresse de la table à transférer est une adresse relative quel que soit le mode de l'appelant. Elle indique un déplacement dans la CDA ou la ZDR dont l'adresse début est contenue dans la mémoire débanalisée '18.

C) Echange Système

C'est l'appel réalisé par le système (mode maître) pour le compte d'un utilisateur. Exemple : chargement d'une tâche non résidente.

Mot -2	Adresse début de la partition		
-1	Octet de Fonction	N° unité d'échange	
A → 0	'FFFF		
1	Adresse de la table à transférer		
2	Compte d'octet à transmettre ou adresse de la table des codes d'arrêt		
3	Compte-rendu d'échange		
4	Evénement fin d'échange		
5	Information complémentaire		Optionnel

L'adresse de la partition est une adresse de format SLO (16 bits de poids forts d'une adresse absolue 20 bits).

Les adresses de l'IOCB et de la table des codes d'arrêt sont des adresses absolues.

L'adresse de la table à transférer est une adresse relative dans la partition.

Attention :

Dans le cas d'IOCB étendu ou de SVC IOCS située dans une autre SVC et dont l'IOCB contient une autre adresse que l'adresse buffer, les mots référencés doivent se trouver dans la même partition (même SLO).



A) Numéro d'unité d'échange

Il indique l'unité sur laquelle porte l'échange. Cette unité peut être une unité symbolique (bit 8 = 1) ou une unité fonctionnelle (bit 8 = 0). Il y a au maximum 128 unités symboliques et 128 unités fonctionnelles. Les numéros d'unités fonctionnelles sont généralement banalisés et peuvent être affectés d'une manière quelconque à la génération.

3 d'entre elles sont interprétées de façon particulière :

- les unités fonctionnelles '00 (ZE), '7F (ME) et '7E (Horloge temps réel) sont débanalisées ; elles ne pointent sur aucune unité, physique.
- toute demande portant sur l'unité fonctionnelle "ZE" est ineffective.

Exemple d'application :

L'assembleur ASM génère une liste du programme assemblé sur l'unité symbolique LO (Liste Output). Si l'utilisateur ne désire pas obtenir une liste il lui suffira de faire pointer l'unité symbolique "LO" sur l'unité fonctionnelle "ZE".

- les seules demandes autorisées portant sur l'unité fonctionnelle '7E (Horloge temps réel) sont les fonctions spéciales de positionnement :

- . Lancement horloge
- . Arrêt horloge

- pour toute demande portant sur l'unité fonctionnelle "ME", IOCS16 ne fait aucun traitement, il fait simplement appel à un sous-programme dont le point d'entrée se trouve dans une mémoire interne au moniteur.

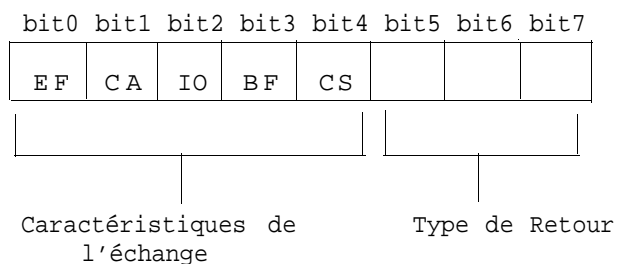
Celle-ci est initialisée par l'utilisateur au moyen d'une SVC (SVC EMAD). Le sous-programme en question trouvera dans le registre A l'adresse de L'IOCB associé à la demande.

L'utilisateur pourra en écrivant le sous-programme de son choix réaliser un traitement particulier à l'unité fonctionnelle "ME".

Exemple d'application :

Rangement en mémoire

B) Octet de fonction



* Caractéristiques de l'échange

- | | |
|--------|---|
| EF = 1 | demande d'échange effectif |
| EF = 0 | fonction spéciale moniteur ou de positionnement |
| CA = 1 | demande avec arrêt sur code |
| CA = 0 | échange sur compte d'octets |
| IO = 1 | demande de sortie (écriture) |
| IO = 0 | demande d'entrée (lecture) |
| BF = 0 | lecture en avant |
| BF = 1 | lecture en arrière |

Remarques :

- BF n'a de sens que si l'échange est une lecture et que le périphérique concerné permet un choix entre la lecture avant et la lecture arrière.
- CS n'a pas de signification générale, il est ignoré d'IOCS16.

Ces deux bits peuvent cependant être utilisés par un driver qui les interprétera comme une "condition spéciale" du transfert.

Exemple CS = 0 les "null", "perfo-on", "Perfo-off" sont ignorés.
CS = 1 les "null", "Perfo-on", "Perfo-off" sont pris en compte.

* Types de retour :

RETOUR EN FIN D'ECHANGE (EMOD) : Fonction codée '0000

- le contrôle est rendu à l'appelant lorsque l'échange est effectivement terminé. Le mot (4) de l'IOCB est utilisé pour positionner un événement fin d'échange.
Cet événement est testé par IOCS16, provoquant alors la suspension du programme appelant jusqu'à la fin de l'échange.
- un compte-rendu du déroulement de l'échange est fourni dans le mot (3) de l'IOCB.

RETOUR APRES INITIALISATION (IMOD) : Fonction codée '0100.

- le contrôle est rendu à l'utilisateur dès que son échange a pu être effectivement initialisé
- il positionne dans le mot (3) de l'IOCB un compte-rendu du déroulement de l'échange.
- il positionne un événement fin d'échange en utilisant le mot (4) de l'IOCB.
Cet événement pourra être testé par la tâche demandant l'échange (ou une autre tâche) grâce à une requête spéciale au superviseur et provoquer la suspension de cette tâche jusqu'à ce que l'échange soit terminé.

RETOUR APRES INITIALISATION AVEC UTILISATION DU POOL BUFFER (IBMOD) :

Fonction Codée '0200

- ce mode de retour n'est utilisable que pour une sortie.
- les utilisateurs disposent d'un certain nombre de buffers (Pool). Le nombre de buffers disponibles et la taille de chacun d'eux sont définis à la génération.
- dans un tel échange IOCS16 transfère l'IOCB et le buffer de l'utilisateur dans un buffer du pool qu'il s'est au préalable réservé. Ce buffer sera libéré à la fin de l'échange.
- IOCS16 rendra la main à l'utilisateur dès que l'échange aura pu être effectivement initialisé ; celui-ci pourra utiliser sa zone de données pour de nouveaux travaux, l'échange se poursuivant entre le buffer alloué et le périphérique.
- Ce mode de retour est incompatible avec :
 - * Une demande d'échange avec buffer situé en CDA.
 - * Une demande d'échange système.

Remarques :

- il n'y a pas positionnement d'un compte-rendu lors d'une telle demande.
Il n'y a pas non plus positionnement d'un événement fin d'échange. Les mots (3) et (4) ne sont pas utilisés.
- la taille de la zone mémoire disponible pour les données (taille d'un buffer du pool, IOCB exclu) est contenue dans la mémoire LBUFP. Elle est fixée à la configuration (macro %POOL). Le compte d'octets à échanger ne devra pas dépasser 2 x (LBUFP).
- le "pool" buffers étant une ressource, s'il n'y a pas de buffer disponible au moment de la demande, la tâche appelante est suspendue jusqu'à la libération d'un buffer.

RETOUR SPECIAL : Fonction codée '0400

- ce mode de retour n'est pas utilisable pour une application en mode esclave.
- il permet de réveiller une application qui est en attente sur un sémaphore paramétré. A la fin de l'échange IOCS16 fait un ACT sur le sémaphore paramétré avec comme paramètre le numéro de la FU où l'échange s'est terminé (voir instruction ACT).
- pour l'application qui a lancé l'échange tout se passe comme pour un échange avec retour immédiat.
- description de l'IOCB

1	C A	I O	B F	C S	1	0	0	N° unité d'échange	mot 0
Adresse de la table à transférer									mot 1
Compte d'octets à transmettre									mot 2
Compte rendu d'échange									mot 3
Adresse du sémaphore privé paramétré									mot 4

C) Adresse de la table à transférer

Elle se trouve dans le mot (1) de l'IOCB. C'est l'adresse du mot contenant le premier octet à transférer. Le premier octet échangé sera toujours l'octet gauche du mot.

D) Echange avec arrêt sur compte d'octets nul

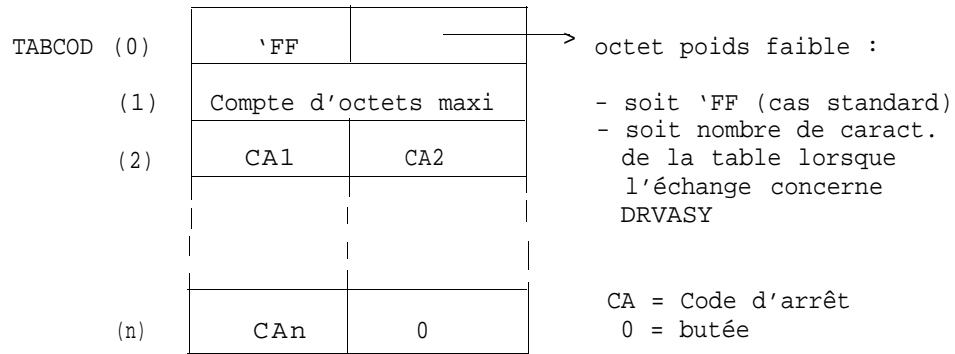
Lorsque le bit CA de l'octet de fonction est à 0, le mot (2) de l'IOCB indique le compte d'octets que l'utilisateur veut échanger avec le périphérique.

Ce nombre est généralement limité à 16 K - 1 octets, cependant il peut être inférieur à cette valeur pour certains périphériques (voir utilisation des drivers).

E) Echange avec arrêt sur code (STOPCD)

* Fonctionnement

- lorsque le bit CA de l'octet de fonction est à 1 le critère d'arrêt de l'échange est la concordance entre un code échangé et l'un des codes d'une table. La table de codes d'arrêt contient un ensemble de codes qui sont tous comparés à tous les caractères échangés ; la concordance de deux caractères provoquera alors la fin de l'échange.
- le mot (2) de l'IOCB contient dans ce cas l'adresse d'une table des codes d'arrêt.
- un compte d'octets maximum est nécessaire pour limiter l'échange en cas de non reconnaissance des codes d'arrêt. Ce compte d'octets peut être :
 - fixé lors de la génération à l'aide de la macro %INPMAX (cf. M..U IOCS16) (valeur par défaut : 80). Dans ce cas la table "TABCOD" ne contient que les codes d'arrêt à raison d'un par octet. Cette table est limitée par un octet nul (0).
 - fourni par l'utilisateur par extension de la table des codes d'arrêt "TABCOD" (mêmes règles que pour le compte d'octets à transmettre) :



L'utilisation d'échanges de ce type est très fréquente sur les périphériques à enregistrement de longueur variable tels que télétypes, lecteurs rapides, elle permet la réalisation de programmes de type dialogue opérateur système. Remarques :

On doit donc dans la rédaction des programmes penser à toujours s'assurer qu'une entrée sur code d'arrêt est effectivement terminée par lecture du compte-rendu. Le compte-rendu permet de connaître le dernier caractère échangé.

* Constitution de la table des codes d'arrêt :

. Exemple avec compte d'octets fixé à la génération (%INPMAX = 256)
TABCOD : BYTE "A", "B", '0

. Exemple avec compte d'octets fixé lors de la demande d'échange (130 octets)

TABCOD : WORD 'FFFF
WORD '82
BYTE "A", "B", '0

dans les 2 cas, les codes d'arrêt sont les caractères A et B.

F) Compte-rendu d'échange

Il est donné en retour du sous-programme superviseur IOCS16, d'une part dans le registre accumulateur et d'autre part dans le mot 3 de l'IOCB.

A = 6000, le mot 3 est sans signification.

L'adresse de la table des paramètres (IOCB) est en dehors de la zone esclave.

A = 0 s'il s'agit d'une fonction spéciale celle-ci a été acceptée. Dans le cas d'un échange effectif avec compte-rendu (fonction '00 ou '01) le mot 3 a la signification suivante :

Il est mis à '8000 lors de la prise en compte de la demande d'échange par IOCS16, il sera modifié en fin d'échange.

La composition de ce mot est alors la suivante :

bit 0 = 1, échange en cours, le reste est sans signification
bit 0 = 0, échange terminé

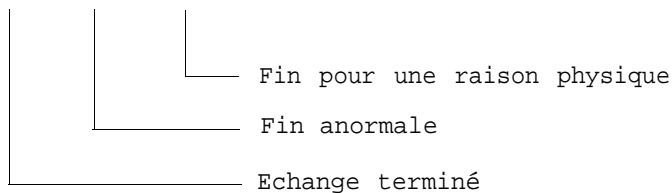
bit 1 = 1, échange terminé anormalement
bit 1 = 0, échange terminé normalement. Les bits 2 à 15 indiquent alors le compte d'octets effectivement échangé.

bit 2 = 1, indique que la demande d'échange était incorrecte
bit 2 = 0, indique que l'échange s'est terminé anormalement pour une cause physique.

Les bits 3 à 15 donnent un code de défaut propre à chaque périphérique lorsque l'échange s'est terminé anormalement pour une cause physique. D'où les 4 types de compte-rendu possibles

DEFAULT PHYSIQUE

bit0	1	2	3
0	1	0	Mot d'état de l'unité physique



L'interprétation du mot d'état de l'unité physique dépend du périphérique sur lequel s'est déroulé l'échange (voir MEM016).

G) Evénement fin d'échange

Lors d'une demande d'échange avec retour après initialisation (fonction '01) ou en fin d'échange (fonction '00), IOCS16 utilise le mot (4) de l'IOCB pour positionner un événement fin d'échange. Ce mot est un sémaphore privé non paramétré (ou de synchronisation).

Dans le cas d'une demande avec retour en fin d'échange, IOCS16 se met en attente de l'événement fin d'échange avant de rendre le contrôle au programme appelant en s'appropriant ce sémaphore pour le compte de la tâche appelante.

Lorsque l'échange demandé sera terminé, la tâche suspendue sera rendue active.

Dans le cas du retour après initialisation l'utilisateur pourra s'il le veut se mettre en attente de la fin d'échange grâce à la fonction WEIO du superviseur (voir chapitre 5). Il peut aussi ne jamais se mettre en attente de l'événement fin d'échange concernant cet IOCB. Le mot (4) devra cependant être présent car il est utilisé par IOCS16.

H) Information complémentaire

Le mot (5) de l'IOCB est ignoré d'IOCS16 mais il peut contenir une information complémentaire du driver

Exemple :

C'est le cas d'une demande d'échange portant sur une unité fonctionnelle disque à accès aléatoire. Le mot (5) fournira au driver l'adresse physique sur le disque.

3.2 FONCTIONS SPECIALES

Ce sont toutes les fonctions qui ne s'accompagnent pas d'un transfert de données. Il existe deux types de fonctions spéciales :

- les fonctions spéciales "moniteur"
- Les fonctions spéciales de "positionnement".

3.2.1 Forme de l'IOCB

Il comporte un seul mot qui se présente sous la forme suivante :

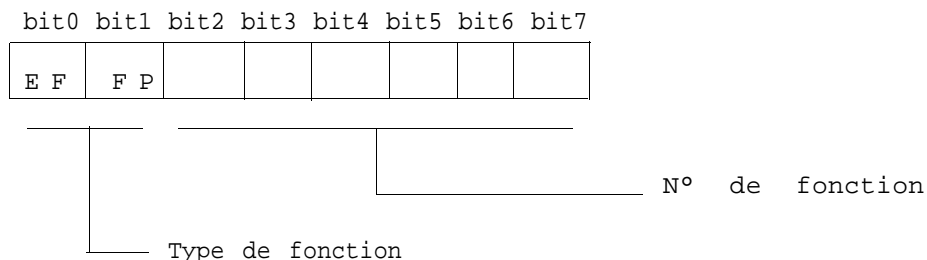
Octet de Fonction	Ne Unité d'échange
-------------------	--------------------

3.2.2 Description de l'IOCB

A) Numéro d'unité d'échange

Il s'agit comme pour une demande d'échange effectif d'un numéro d'unité symbolique ou d'unité fonctionnelle.

B) Octet de fonction



EF = 0 Caractérise une fonction spéciale

FP = 1 Caractérise une fonction spéciale de positionnement

FP = 0 Caractérise une fonction spéciale moniteur.

* Fonctions spéciales de positionnement

Elles concernent les fonctions particulières à un périphérique donné qui ne sont pas associées à un échange de données et sont réalisées au niveau du driver.

Chaque driver dispose d'une entrée spéciale pour les fonctions de positionnement. Le numéro de la fonction demandée est interprété d'une façon particulière pour chaque driver.

(Se reporter aux manuels d'exploitation particuliers à chaque périphérique).

Exemple :

Sur le dérouleur de bande, les fonctions de positionnement sont les suivantes :

Valeur associée à la fonction	Code mnémorique	Fonction
'4900	WEOF	Ecriture EOF
'4A00	WGAP	Création d'un GAP
'4B00	BSRCD	Saut arrière de 1 bloc
'4C00	FSRCD	Saut avant de 1 bloc
'4D00	BSFILE	Saut arrière de 1 fichier
'4E00	FSFILE	Saut avant de 1 fichier
'4800	RWD	Rebobinage
'4700	RWDU	Rebobinage et passage en local

Remarque :

- Le retour à l'utilisateur se fait toujours après initialisation de la fonction de positionnement
- l'utilisateur n'a pas de contrôle sur le déroulement d'une fonction de positionnement (pas de compte-rendu) en particulier il n'est pas averti de la fin d'exécution de cette fonction.
- cependant l'unité physique concernée par la fonction reste "occupée logiquement" jusqu'à la fin d'exécution de la fonction. Toute demande portant sur cette unité physique (demande d'échange effectif ou de fonction spéciale) sera donc mise en attente pendant le déroulement de la fonction.

* Fonctions spéciales adressées au moniteur

- elles concernent les fonctions d'intérêt général, c'est-à-dire, celles qui ne sont pas caractéristiques d'un périphérique.
- elles ne sont jamais associées à un échange d'information et sont exécutées au niveau du moniteur.
- elles sont exécutées immédiatement : il n'y a pas d'occupation logique d'unité physique.

Un certain nombre de ces fonctions sont standard mais un utilisateur peut créer très facilement de nouvelles fonctions. En effet, chacune de ces fonctions peut être exécutée par un module dont IOCS16 trouve le point d'entrée dans une table (TBFSM) qu'il gère. Ces nouvelles fonctions peuvent très facilement être introduites au moment de la génération d'IOCS16 (voir la macro %FSMON dans le manuel d'utilisation IOCS16).

Les fonctions "standard" sont les suivantes :

Valeur associée à la fonction	Code mnémorique	Fonction
'0000	CLEAR	Réinitialisation du système
'0100	PUSI	Lecture mot d'état de l'unité physique (PU)
'0200	PUA	Attachement de la PU au niveau Soft demandeur
'0300	PUD	Détachement de la PU au niveau Soft demandeur
'0600	KILL	Mettre fin à un échange
		Initialiser une PU
'0800	FDMONT	Démontage d'un volume (cf. § 7.3.1)
'0900	FMONT	Montage d'un volume (cf § 7.3.2)
'0A00	FREAD	Lecture de la structure d'un volume (cf. § 7.3.3)

Remarques :

- 1) les fonctions concernent généralement une unité physique. Cependant, la demande à IOCS16 portera toujours sur une unité symbolique ou une unité fonctionnelle liée à l'unité physique en question, elles seront ignorées si elles portent sur l'unité fonctionnelle ZE (valeur '0000)
- 2) les fonctions d'attachement, détachement annulation de demande d'échange et d'initialisation d'unité physique n'existent que dans la version IOCS16 disponible sur machine avec gestion des tables et ressources par scheduler et sémaphores microprogrammés.

FONCTION DE REINITIALISATION DU SYSTEME (CLEAR)

Cette fonction est toujours acceptée et exécutée immédiatement ; elle a le rôle suivant :

- elle libère toutes les ressources du système (unités physiques, buffers du pool, canaux)
- elle annule toutes les fonctions d'attachement ou d'échange
- elle initialise les PST des niveaux d'entrées-sorties
- elle initialise les mémoires débanalisées (relais des boites à lettres et micro-canaux)
- elle initialise les mémoires internes à IOCS16
- elle initialise le masque des interruptions (IM) en fonction des périphériques gérés.

FONCTION DE LECTURE DU MOT D'ETAT D'UNE UNITE PHYSIQUE (PUSI)

Cette fonction est toujours acceptée et exécutée immédiatement. Elle a pour rôle de fournir à l'utilisateur les informations relatives à l'unité physique pointée par l'unité fonctionnelle (ou symbolique) sur laquelle porte la demande :

- dans le registre A, le mot d'état de l'unité physique (cf MEMO16)
- dans le registre X, le mot 0 (STATUS) de la table d'unité physique (voir manuel d'utilisation).

Dans Le cas où la demande porte sur une unité physique inconnue, le registre A est rendu égal à '6000.

FONCTIONS D'ATTACHEMENT ET DE DETACHEMENT D'UNE UNITE PHYSIQUE (PUA et PUD)

On rappelle que les demandes à IOCS16, bien que s'adressant à une unité symbolique ou fonctionnelle sont gérées au niveau des unités physiques, une unité physique donnée étant généralement attribuée au demandeur pour la durée d'un échange (ou d'une fonction de positionnement).

Par la fonction d'attachement, une tâche a la possibilité de s'approprier une unité physique non plus pour la durée d'un échange, mais tant qu'elle jugera nécessaire.

En effet si un utilisateur s'est "attaché" une unité physique donnée, toutes les demandes en provenance d'une autre tâche, adressées à cette unité physique, seront mises en attente tant qu'il n'aura pas décidé de la détacher par une fonction de détachement.

Exemple d'utilisation de ces fonctions :

Soient deux tâches T1 et T2 ayant à fournir sur des unités symboliques qui peuvent être différentes deux journaux J1 et J2.

La configuration du système d'entrées-sorties peut être telle que ces deux unités symboliques soient associées à la même unité physique, une imprimante par exemple.

Pour la clarté de l'impression des journaux, il serait souhaitable que des échanges demandés par T2 et destinés au journal J2 ne viennent pas s'intercaler entre deux échanges demandés par T1 et destinés au journal J2 (et réciproquement). Les deux tâches résoudre ce problème en demandant avant de commencer l'impression de leur journal, L'attachement de l'unité physique associée à l'unité symbolique sur laquelle leur journal est imprimé.

Une fois leur journal imprimé, elles devront évidemment détacher l'unité physique.

Règles générales de fonctionnement.

- 1) Lorsqu'une tâche demande l'attachement d'une unité physique, sa demande ne sera satisfaite que si l'unité physique en question est disponible. Si celle-ci ne l'est pas, soit parce que l'unité physique est en cours d'échange, soit parce qu'elle a déjà été attachée à une autre tâche, le programme appelant est suspendu jusqu'à ce que l'unité physique qu'il veut s'attacher soit disponible.

Reprenons l'exemple précédent :

Les tâches T1 et T2 ont les séquences de programme suivantes :

T1	T2
.	.
.	.
Demande d'attachement portant sur SU1	Demande d'attachement portant sur SU2
.	.
.	.
.	.
Echanges pour impression de J1	Echanges pour impression de J2
.	.
.	.
.	.
Détachement de SU1	Détachement de SU2
.	.
.	.

Supposons que les unités symboliques SU1 et SU2 concernées par les journaux J1 et J2 soient associées à la même unité physique.

Supposons d'autre part que pour une cause quelconque T1 soit rendue active après la demande d'attachement faite par T1. La demande d'attachement de T2 provoquera sa suspension jusqu'à ce que T1 détache l'unité physique.

- 2) lorsqu'un programme demande le détachement d'une unité physique qui lui est attachée, IOCS16 n'exécute la demande que lorsque l'unité physique n'est plus en cours d'échange.

Sinon, le programme qui a demandé le détachement est suspendu jusqu'à la fin de l'échange en cours.

Remarques :

Le retour à l'utilisateur se fera avec :

A = 0 dans le cas où la fonction a pu être exécutée normalement.

A = '4000 dans les cas suivants :

- . Demande d'attachement à une unité physique déjà attachée au demandeur.
- . Demande de détachement d'une unité physique qui n'est pas attachée au demandeur.

A = '6000 si l'unité d'échange n'est pas gérée par IOCS16.

FONCTION ARRET D'UN ECHANGE (KILL)

Cette fonction permet d'interrompre un échange et de libérer les ressources qui avaient été allouées pour cet échange :

- l'unité physique ou périphérique sur lequel portait l'échange
- la tâche qui avait demandé l'échange (uniquement dans le cas de retour en EMOD) sera "réveillée".

Remarque :

Le retour à l'utilisateur se fera avec :

A = 0 dans le cas où la fonction a pu être exécutée normalement

A = '6000 dans le cas où l'unité d'échange n'est pas gérée par IOCS16

FONCTION CLEAR SELECTIF (SCLEAR)

Cette fonction a le rôle suivant :

Elle annule toutes les demandes d'échange toutes les fonctions d'attachement sur l'unité d'échange précisée dans l'IOCB.

Remarque :

Le retour à l'utilisateur se fera avec :

A = 0 dans le cas où la fonction a pu être exécutée normalement

A = '6000 dans le cas où l'unité d'échange n'est pas gérée par IOCS16.

4 ECRITURE SYMBOLIQUE DE L'IOCB

Pour faciliter l'écriture symbolique des demandes à IOCS16, on recommande l'utilisation d'un certain nombre de codes numériques décrivant les demandes.

4.1 DEMANDE D'ECHANGE EFFECTIF

4.1.1 Mot de fonction

Code Héxadécimal	Code Mnémonique	Fonction Réalisée
'4000	CA	Echange sur code d'arrêt. Par défaut l'échange sera un compte d'octets
'2000	OUT	Sortie (OUTPUT)
'0000	IN	Entrée (INPUT)
'1000	INBCK	Lecture arrière (BACKWARD)
'0800	NSN	Pas de suppression des NULL en lecture (NOSUPRESS NULL)
'0800	NVERIF	Pas de vérification ni de réessai en lecture ou écriture disque

4.1.2 Type de retour à l'utilisateur

Code Héxadécimal	Code Mnémonique	Fonction réalisée
'8000	EMOD	Retour en fin d'échange
'8100	IMOD	Retour immédiat après initialisation
'8200	IBMOD	Retour immédiat et utilisation du
'8400	IRMOD	Retour Spécial

Code Héxadécimal	Code Mnémorique	Fonction réalisée
'0000	CLEAR	Réinitialisation du système (uniquement en mode maître)
'0100	PUSI	Lecture du mot d'état [physical unit status input)
'0200	PUA	Attachement d'une unité physique (physical unit attach)
'0300	PUD	Détachement d'une unité physique (physical unit detach)
'0600	KILL	Mettre fin à un échange
'0700	SCLEAR	Réinitialisation d'une unité physique
'0800	FDMONT	Démontage d'un volume
'0900	FMONT	Montage d'un volume
'0A00	FREAD	Lecture de la structure d'un volume

4.3 FONCTIONS SPECIALES DE POSITIONNEMENT

Gérées par les drivers leur description se trouve dans les manuels d'exploitation des périphériques.

5 LA FONCTION ATTENTE DE FIN D'OPERATION D'ENTREE-SORTIE : WEIO

Cette fonction du superviseur qui est liée au moniteur d'entrées-sorties IOCS16 permet à une tâche software de se mettre en attente d'une fin d'opération d'entrée-sortie effective.

Elle utilise le mot (4) de la table de contrôle de la demande d'entrée-sortie.

Si le transfert n'est pas terminé lors de la demande, la tâche se trouve alors suspendue jusqu'à ce que l'événement fin d'échange du transfert considéré arrive.

Cet événement réactivera la tâche qui retrouvera le contrôle en séquence, et mettra à jour le compte-rendu dans l'IOCB.

Si le transfert est terminé lors de la demande, le moniteur rend immédiatement le contrôle en séquence à la tâche appelante.

Remarque :

Une seule tâche peut se mettre en attente de fin d'opération d'entrée-sortie mais celle-ci n'est pas obligatoirement celle qui a demandé l'échange.

Séquence d'appel :

LAD	IOCB1
SVC	WEIO

- où IOCB1 est l'adresse de la table de contrôle de l'opération d'entrée-sortie d'une demande d'échange effectif préalable qui avait été faite avec retour après initialisation (IMOD).
- WEIO désigne le sous-programme moniteur qui traite cette fonction.
- le sous-programme WEIO est réentrant
- il se déroule sous niveau de priorité de la tâche appelante et utilise la zone K de l'utilisateur
- il utilise le mot (4) de la table de contrôle associée à l'échange dont on veut attendre la fin.

6 A C T I V A T I O N D E L A T A C H E A L A R M E

IOCS16 active la TACHE ALARME (tâche hardware système de priorité 0) au moment de l'instruction ACTD (Activate Debug) dans les cas, et avec les paramètres suivants :

- 1 = Interruption inattendue (niveau ou sous-niveau hardware non géré).
- 3 = Défaut dans le fonctionnement du canal (non réponse à une instruction IPI d'initialisation ou libération du canal).
- 4 = Appel opérateur reçu 4 fois consécutives.

7 GESTION DU MONTAGE DE VOLUME

7.1 GENERALITES

Le logiciel de gestion du montage de volume se présente sous la forme de trois fonctions spéciales moniteur intégrées.

L'utilisation de ce logiciel sous les superviseurs permet une meilleure gestion des disques par un accroissement de la sécurité et une utilisation plus souple de ces supports. L'augmentation de ces performances est obtenue par :

- la configuration dynamique des tables de gestion du disque à partir de la structure définie sur le support disque.
- la gestion des espaces disques (espace ouvert à un instant donné, accessible en lecture et écriture) d'où un renforcement de la fiabilité,
- la prise en compte automatique, des cylindres en défaut,
- la portabilité des supports amovibles d'une installation à l'autre,
- la portabilité des programmes,
- une plus grande souplesse de génération.

L'ensemble de ces services est directement accessible au moyen de commandes opérateur simples offertes par les différents systèmes d'exploitation. Il comprend deux niveaux de service :

- LA GESTION DES SUPPORTS : gestion des labels et des cylindres en défaut,
- LA GESTION D' ESPACES structuration des supports disque et reconfiguration dynamique des tables d'IOCS16.

Attention : Lorsqu'un système possède l'option de gestion d'espace toutes les unités physiques fonctionnent avec ce service et tous les supports disques correspondants doivent être structurés en espaces (commande SDEF de FUP4).

La commande IDEF de FUP4 permet de rendre exploitables les supports disques traités auparavant sans gestion d'espace, sans être obligé de les reformater.

7.2.1 Définitions

On appelle :

- Unité physique : un périphérique électro-mécanique (exemple : unité de disque raccordée à la voie 2 d'un coupleur).
- Support : une mémoire magnétique (bande, disque etc...).
- Support fixe : un support fixé physiquement à une unité physique par construction.
- Support amovible : un support qui peut être monté et démonté sur une unité physique.
- Volume : un support identifié par un label.
- Espace : une partie d'un support définie par son adresse physique et sa longueur. Il est identifié par son numéro (de zéro à 27).
- FU ouverte : une unité fonctionnelle disque est dite ouverte lorsqu'il lui correspond dans les tables d'IOCS16 un couple (adresse, longueur) définissant un espace disque. Dans le cas contraire une FU est dite fermée.

7.2.2 Principes de reconfiguration dynamique des tables de gestion du disque

A) La table d'espaces d'un support

La structure d'un support est décrite dans sa table d'espaces. La table d'espaces du support est située sur le secteur 4 du cylindre zéro du disque. Elle est initialisée par le programme de formatage et mise à jour à la création des espaces.

Le premier espace décrit sur la table d'espaces est appelé ESPACE INITIAL. Il est défini par le formatage. Il débute sur le cylindre 0 du support. Sa longueur est non nulle, normalement égale au nombre de cylindres formatés.

Les espaces suivants sont créés par la commande SDEF de FUP4.

La commande IDEF de FUP4 permet de créer l'espace initial et d'initialiser la table des espaces pour rendre exploitables les supports disques traités auparavant sans gestion d'espace. sans être obligé de les reformater.

B) Les tables de gestion du disque

Les tables de gestion du disque sont décrites dans les tables d'IOCS16. Elles mémorisent le découpage d'une unité physique en unités fonctionnelles.

Ce découpage est défini à la génération du système par les macro-instructions de GI016.

Une unité physique est identifiée par une unité fonctionnelle particulière appelée FU INITIALE.

Cette unité fonctionnelle présente les caractéristiques suivantes :

- elle est la première décrite sur l'unité physique lors de la génération (macro-instruction de GI016),
- elle est toujours accessible en lecture,
- elle appartient à la plage des FU mnémoniques D1 à DF, E1 à EF (sauf EC).

Les unités fonctionnelles suivantes peuvent être des FU mnémoniques OU des FU numériques sans nom (de 58 à 125).

C) Principe et règle de la reconfiguration dynamique

La reconfiguration dynamique des tables d'IOCS16 consiste à charger les tables de gestion du disque avec les valeurs inscrites sur le support dans la table d'espaces.

Cette opération affecte à chaque unité fonctionnelle disque un espace disque et lui attribue la valeur des paramètres (adresse, longueur) écrits sur le support en respectant la règle d'affectation par rang.



Table d'IOCS16

Support disque - Table d'espaces

Les unités fonctionnelles Di, Dj, Dk sont générées sur la même unité physique.

La reconfiguration dynamique des tables d'IOCS16 se fait suivant la règle suivante :

LE Ieme ESPACE D'UN SUPPORT EST ACCESSIBLE PAR LA Ieme
UNITE FONCTIONNELLE DE L'UNITE PHYSIQUE

L'ordre d'affectation des unités fonctionnelles sur l'unité physique est défini par l'ordre de génération. Cet ordre ne suit pas l'ordre croissant des numéros des unités fonctionnelles mais l'ordre de description des unités fonctionnelles disque au fur et à mesure de leur génération.

La reconfiguration dynamique est exécutée par les commandes de montage de volume (ex. : MONT de BOS16) ou automatiquement à l'initialisation des systèmes (ex : TSM16).

3 Gestion d'espace standard

Elle nécessite l'utilisation de disque structuré en espaces et la reconfiguration dynamique.

Programmation portable sur SU

Afin de satisfaire au critère de portabilité des programmes et des supports la gestion d'espace standard favorise la programmation sur unités symboliques (SU). En effet les associations SVC E/S --> SU et SU --> Nième espace disque sont indépendantes des noms de FU définis à la génération du système d'exploitation.

L'association SVC E/S --> SU se fait par programme et l'association SU --> Nième espace disque par commande.

Exemple :

La génération des tables d'IOCS16 se fait par la méthode dite GENERATION DYNAMIQUE STANDARD d'une unité physique.

Dans ce type de génération l'adresse et la taille de ces unités fonctionnelles définies à la génération du système sont quelconques puisqu'elles seront mises à jour au fur et à mesure de la reconfiguration dynamique sauf pour la FU initiale qui doit être générée ouverte : adresse = 0. et longueur ≠ 0 afin de pouvoir accéder au cylindre 0.

Il est recommandé de générer la FU initiale avec les paramètres définissant l'espace initial : adresse = 0 et longueur = nombre de cylindres du support, et de générer fermées les FU suivantes : adresse = 0 et longueur = 0 afin d'en interdire l'accès avant d'avoir effectué les opérations de montage de volume.

Soit une unité physique sur laquelle ont été générées, dans cet ordre, les unités fonctionnelles disques suivantes :

- D9 (adresse = 0, longueur = 400) FU initiale
- D3 (adresse = 0, longueur = 0)
- D6 (adresse = 0, longueur = 0).

Soit un support structuré en 2 espaces disques dont les caractéristiques sont les suivantes :

- ESPACE 0 (adresse = 0, longueur = 400) ESPACE INITIAL
- ESPACE 1 (adresse = 1, longueur = 399).

La RECONFIGURATION DYNAMIQUE des tables d'IOCS16 réalise les affectations suivantes :

- D9 <--> ESPACE 0 (@ = 0, L = 400)
- D3 <--> ESPACE 1 (@ = 1, L = 399)
- D6 FU fermée

Pour un autre support disque structuré en 3 espaces disque avec les caractéristiques suivantes :

- ESPACE 0 (adresse = 0, longueur = 400)
- ESPACE 1 (adresse = 1, longueur = 300)
- ESPACE 2 (adresse = 301, longueur = 99).

La reconfiguration dynamique des tables d'IOCS16 réalise les affectations suivantes :

- D9 <--> ESPACE 0 (@ = 0, L = 400)
- D3 <--> ESPACE 1 (@ = 1, L = 300)
- D6 <--> ESPACE 2 (@ = 301, L = 99).

Programmation figée sur FU

La programmation sur FU fige par convention la correspondance SV E/S --> FU --> (adresse, longueur).

Il existe une possibilité de récupérer la programmation des supports désignés par FU. Il suffit de fabriquer les tables d'espaces des supports de façon à s'adapter au système généré : cela est possible en décrivant des "trous" dans la table d'espaces afin de définir toutes les correspondances FU <--> (adresse, longueur).

Exemple :

Soit un système comportant une unité physique sur laquelle ont été générées les unités fonctionnelles D9, D3, D5, D6, E1, E2 par la méthode de la "génération dynamique standard".

- D9 (adresse = 0, longueur = 400) FU initiale
- D3 (adresse = 0, longueur = 0)
- D5 (adresse = 0, longueur = 0)
- D6 (adresse = 0, longueur = 0)
- E1 (adresse = 0, longueur = 0)
- E2 (adresse = 0, longueur = 0).

- A) Supposons que l'on veuille reconfigurer les tables d'IOCS16 de façon à ce que les unités fonctionnelles E1 et E2 soient accessibles lors de la prise en compte du support par IOCS16, il faut décrire les espaces 1, 2, 3 comme des espaces vides.

```
ESPACE 0 (adresse = 0, longueur = 400) Espace initial
ESPACE 1 (adresse = 0, longueur = 0) Espace vide
ESPACE 2 (adresse = 0, longueur = 0) Espace vide
ESPACE 3 (adresse = 0, longueur = 0) Espace vide
ESPACE 4 (adresse = 1, longueur = 2).
ESPACE 5 [adresse = 3, longueur = 397).
```

La reconfiguration dynamique des tables d'IOCS16 réalise les affectations suivantes :

```
D9 <----- > ESPACE 0 (@ = 0, L = 400)
D3 FU fermée ESPACE 1 (@ = 0, L = 0)
D5 FU fermée ESPACE 2 (@ = 0, L = 0)
D6 FU fermée ESPACE 3 (@ = 0, L = 0)
E1 <----- > ESPACE 4 (@ = 1, L = 2)
E2 <----- > ESPACE 5 (@ = 3, L = 397).
```

- B) Supposons que l'on se propose à présent de gérer les unités fonctionnelles D5 et D6 à partir de ce même système. Il faut décrire les espaces 1, 4, 5 comme des espaces vides. Après la reconfiguration dynamique des tables d'IOCS16 on obtient les affectations suivantes :

```
D9 <----- > ESPACE 0 (@ = 0, L = 400)
D3 FU fermée ESPACE 1 (@ = 0, L = 0)
D5 <----- > ESPACE 2 (@ = 1, L = 300)
D6 <----- > ESPACE 3 (@ = 301, L = 99)
E1 FU fermée ESPACE 4 (@ = 0, L = 0)
E2 FU fermée ESPACE 5 (@ = 0, L = 0)
```

7.2.4 Gestion statique des disques

Dans ce type de gestion des disques les FU définies sont directement exploitables sans reconfiguration dynamique. L'adresse et la taille des unités fonctionnelles doivent être générées en fonction de la structure des supports disques correspondants.

Ce type de gestion doit être utilisé pour :

- A) les disques sans gestion d'espaces.

Dans ce cas les supports disques ne sont que formatés et l'association FU <--> (adresse, longueur) est figée à la génération.

- B) les disques avec gestion d'espace mais sans reconfiguration (ex : le disque système sous BOS16)

Ce type de génération est nécessaire quand les FU doivent être initialisées ouvertes et opérationnelles.

Exemple de génération :

D1 (adresse = 0, longueur = 40) FU initiale

D2 (adresse = 40, longueur = 242)

D3 (adresse = 282, longueur = 118)

Pour que les FUP fonctionnent le support doit être structuré en accord avec la génération par l'intermédiaire de la commande SDEF de FUP4 où la FU initiale est la FU D1.

ESPACE 0 (adresse = 0, longueur = 40) Espace initial

ESPACE 1 (adresse = 40, longueur = 242)

ESPACE 2 (adresse = 282, longueur = 118)

7.2.5 Gestion d'espace préinitialisée

Ce type de gestion est utilisé pour les disques avec gestion d'espace et sans reconfiguration dynamique qui nécessite que les FU soient initialisées ouvertes et opérationnelles. Elle permet d'avoir des macros de génération compatibles entre différents systèmes.

Exemple de génération :

- D7 (adresse = 0, longueur = 400) FU initiale

- D1 (adresse = 0, longueur = 40)

- D2 (adresse = 40, longueur = 242)

- D4 (adresse = 282, longueur = 110)

- D8 (adresse = 0, longueur = 0)

D1, D2, D4 sont définis en statique car ils sont utilisés sous BOS16.

D8 est défini en dynamique standard car il est utilisé sous TSM16.

Pour que les FUP fonctionnent le support doit être structuré en accord avec la génération pour les espaces gérés par BOS16.

Exemple :

- ESPACE 0 (adresse = 0, longueur = 400) Espace initial

- ESPACE 1 (adresse = 0, longueur = 40)

- ESPACE 2 (adresse = 40, longueur = 242)

- ESPACE 3 (adresse = 282, longueur = 110)

- ESPACE 4 (adresse = 392, longueur = 81)

Schéma de la reconfiguration dynamique :

L'affectation FU <--> (adresse, longueur) est figée à la génération sous BOS16 et reconfigurable sous TSM16.

- D7 <====> ESPACE 0 (@ = 0, L=400)

- D1 <====> ESPACE 1 (@ = 0, L = 40)

- D2 <====> ESPACE 2 (@ = 40, L = 242)

- D4 <====> ESPACE 3 (@ = 282, L = 110)

- D8 FU fermée

7.3 DESCRIPTION

Les fonctions spéciales moniteur assurant la gestion du montage des volumes sont les suivantes :

Valeur associée à la fonction	Commande mnémonique	Fonction
'8000	FDMONT	Démontage du volume (requête système)
'0900	FMONT	Montage d'un volume (requête système)
0A00	FREAD	Lecture de la structure d'un volume (requête utilisateur)

7.3.1 Requête de démontage

FDMONT

Cette requête est réservée aux systèmes d'exploitation.

A) Séquence d'appel en assembleur :

LAD	IOCB
SVC	IOCS

Cette requête est INTERDITE en mode ESCLAVE

B) Forme de l'IOCB

Mot	N° FONCTION	N° UNITE D'ECHANGE
0		
1	RESERVE	
2	RESERVE	
3	COMPTE-RENDU	
4	RESERVE	
5	RESERVE	

* Numéro de Fonction : '08

* Numéro d'unité d'échange : Numéro de SU/FU. Dans le cas d'un numéro de FU, il s'agit de l'une des FU générées sur l'unité physique (en principe il s'agit de la FU initiale).

* Compte-rendu : il est donné en retour du sous-programme d'une part dans le registre accumulateur et d'autre part dans le mot 3 de l'IOCB.

A = '6000 : le mot 3 est sans signification
. Fonction spéciale moniteur absente (n'est pas générée, au niveau GIO16
. Unité d'échange non gérée
. Appelant en mode esclave.

A = 1 : le mot 3 peut prendre alors les valeurs suivantes :
. '6001 : il n'y a pas de gestion de volume sur ce disque.
. 0 : la requête s'est terminée normalement.

Remarque : la requête est inefficace si le volume est déjà démonté.

C) Fonctions réalisées :

- . Contrôle de la validité des paramètres de l'IOCB : fonction et unité d'échange gérées.
- . Ferme l'ensemble des unités fonctionnelles appartenant à l'unité physique. Toute demande d'échange ultérieure sur l'une d'elles provoquera l'erreur :
'4080 = "zone adressée inexistante"
- . Contrôle la nature de l'appelant (refus de la requête en mode esclave).

7.3.2 Requête de montage

FMONT

Cette requête est réservée aux systèmes d'exploitation.

A) Séquence d'appel en assembleur :

LAD	IOCB
SVC	IOCS

Cette requête est INTERDITE en mode ESCLAVE.

B) Forme de l'IOCB

Mot	N° FONCTION	N° UNITE D'ECHANGE
0		
1	ADRESSE BUFFER	
2	RESERVE	
3	COMPTE-RENDU	
4	RESERVE	
5	RESERVE	

* Numéro de Fonction : '09

- * Numéro d'unité d'échange : Numéro de SU/FU. Dans le cas d'un numéro de FU, il s'agit de l'une des FU générées sur l'unité physique (en principe il s'agit de la FU initiale).
- * Adresse buffer : adresse d'un buffer de longueur égale à 128 mots. Il contient au moment de la demande le label du disque à monter.



Le label se compose de n caractères ($1 \leq n \leq 7$) le n + lieme caractère étant un CR (code ASCII = '8D).

Dans le cas où $n < 7$ les caractères non présents sont complétés par des zéros.

- * Compte-rendu : il est donné en retour du sous-programme d'une part dans le registre accumulateur et d'autre part dans le mot 3 de l'IOCB.
 - A= '6000 : le mot 3 est sans signification
 - . Fonction spéciale moniteur absente (n'est pas générée au niveau GI016)
 - . Unité d'échange non gérée
 - Appelant en mode esclave.
 - A = 1 : le mot 3 peut prendre alors les significations suivantes :
 - '6001 : il n'y a pas de gestion de volume sur l'unité physique ou sur le support à monter.
 - '6002 : les espaces du support précédent ne sont pas fermés.
 - '6003 : le label n'est pas celui en cours sur le support.
 - '6004 : le système généré n'est pas conforme (ne dispose pas d'unité fonctionnelle initiale).
 - '6005 : il y a plus d'espaces décrits sur le support qu'IOCS16 ne peut en gérer. Dans ce cas les premiers sont pris en compte.
 - '6006 : le volume n'est pas structuré (secteur 3 ou 4 non conforme).
 - '6007 : il n'y a pas de cylindre valide parmi les huit premiers du support.
 - '4xxx : indique que l'échange s'est terminé anormalement pour une cause physique.
 - 0 : la requête s'est terminée normalement, le contenu du buffer est alors le suivant :
- A = 1 et mot 3 = 0
- . Le buffer contient la table d'espace du support mis à jour par des informations
 - Cf : compte-rendu de la requête FREAD quand RA = 1 et mot 3 = 0.

C) Fonctions réalisées :

- . Contrôle de la validité des paramètres de l'IOCB : fonction et unité d'échange gérées.
- . Contrôle de la nature de l'appelant (refus de la requête en mode esclave).
- . Vérifie l'identité de label entre celui communiqué par la requête et celui du support en cours.
- . Contrôle de la validité de la table des espaces.
- . Mise à jour de la liste des cylindres en défaut de l'unité physique à partir du support.
- . Mise à jour des tables de gestion du disque relatives à l'unité physique : ouverture des unités fonctionnelles.

7.3.3 Requête d'identification d'un support :

FREAD

L'appel de cette requête est possible aux processeurs système et programmes utilisateur.

A) Séquence d'appel en assembleur :

LAD	IOCB
SVC	IOCS

Cette requête est ACCESSIBLE quel que soit le mode de l'appelant.

B) Forme de l'IOCB

Mot	N° FONCTION	N° UNITE D'ECHANGE
0		
1	ADRESSE BUFFER	
2	RESERVE	
3	COMPTE-RENDU	
4	RESERVE	
5	RESERVE	

* Numéro de Fonction : 'A

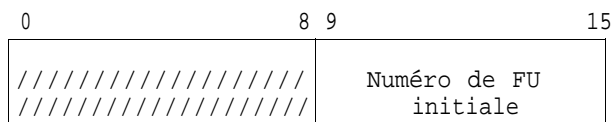
* Numéro d'unité d'échange : Numéro de SU/FU. Dans le cas d'un numéro de FU, il s'agit de l'une des FU générées sur l'unité physique (en principe il s'agit de la FU initiale).

* Adresse buffer : adresse d'un buffer de longueur égale à 128 mots.

* Compte-rendu : il est donné en retour du sous-programme d'une part dans le registre accumulateur et d'autre part dans le mot 3 de l'IOCB.

A = '6000 : le mot 3 est sans signification
 . Fonction spéciale moniteur absente (n'est pas générée, au niveau GI016).
 . Unité d'échange non gérée
 Adresse IOCB est en dehors de la zone esclave.

A = 1 : le mot 3 peut prendre alors les significations suivantes :
 '6000 : le buffer est situé hors de la zone esclave.
 '6001 : il n'y a pas de gestion de volume sur l'unité physique ou sur le support à identifier.
 '6002 : il n'y a pas de volume monté sur cette unité physique. Le buffer contient alors le secteur 3 et le mot 4 de l'IOCB le numéro de FU initialise (voir erreur '6006).
 '6004 : le système généré n'est pas conforme (ne dispose pas d'unité fonctionnelle initiale).
 '6006 : volume non structuré. Le buffer contient alors le secteur 3 et le mot 4 de l'IOCB le numéro de FU initiale :



Description du buffer dans le cas où le mot 3 de IOCB prend une des valeurs : '6002, '6004, '6006.

Il contient le secteur 3 du support. (Voir page suivante).

0	
1	TABLE DES
2	
3	CYLINDRES
4	
5	EN DEFAULT
6	
7	
8	'89AB
9	
10	LABEL
11	
12	
13	CODE STRUCTURE
14	
15	
16	NOM DU FICHIER
17	DU SYSTEME 0
18	
37	NOM DU FICHIER
38	DU SYSTEME 7
39	
127	TYPE CARTOUCHE

CODE STRUCTURE =

- 0 - volume non structuré
- 1 - disque CDB10
- 2 - " DPB50
- 3 -
- 4 - disque CDB20 partie fixe
- 5 - " " " mobile
- 6 - disque souple double face, double densité
- 7 - disque à interface SMD
- 8 - disque WINCHESTER

RA = 1 et MOT 3 = 0 : la requête FREAD s'est terminée normalement. Le buffer contient la table d'espace disque du support mis à jour par des informations de la génération. (= secteur 4 du support).

0			
1	LABEL		
2			
3			
4	CODE SUPPORT		
5	TAILLE SECTEUR		
6	TAILLE CYLINDRE		
7	NBESP	NBFU*	
8	NBFMS		
9			
10	LIBRES		
11			
12	F U *	INDIC	
13	LTAG		ESPACE 0 (espace initial)
14	ADRESSE DISQUE		
15	LONGUEUR		
120	F U *	INDIC	
121	LTAG		ESPACE 27
122	ADRESSE DISQUE		
123	LONGUEUR		
124			
125			
126	SANS SIGNIFICATION		
127			

* : mis à jour à partir des tables d'IOCS16 générées. Le reste des informations provient de la table d'espace disque.

Signification des différentes valeurs

1) - Paramètres communs à l'ensemble des espaces

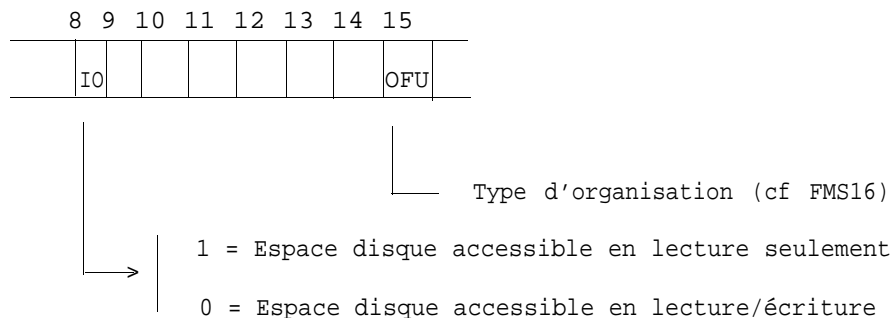
LABEL : de 1 à 7 caractères ASCII, plus RC
CODE SUPPORT :
1 → disque CDB10
2 → disque DPB50
4 → disque CDB20 partie fixe
5 → disque CDB20 partie mobile
6 → disque souple double face double densité
7 → disque à interface SMD
8 → disque WINCHESTER

TAILLE SECTEUR: 128 ≤ TS ≤ 2048 mots de plus TS = 2**11
TAILLE CYLINDRE : 1 ≤ TC ≤ 2**15 - 1 secteurs
NBESP : nombre d'espaces valides décrits sur le support
NBFU : nombre d'unités fonctionnelles configurées par
GI016 pour l'unité physique (1 NBFU 27).
NBFMS: nombre d'espaces disque gérés par FMS (cf notice
FMS16).

2) - Paramètres relatifs à un espace

FU : numéro d'unité fonctionnelle affecté par IOCS16 à
l'espace disque : la nième FU est affectée au nième
espace .

INDIC :



LTAG : Longueur de la Taille d'Allocation de Granule. Si
LTAG = 0 l'espace disque n'est pas géré par FMS16.
ADRESSE DISQUE : Adresse physique, exprimée en cylindres, de
l'espace.
LONGUEUR : Taille de l'espace disque exprimée en cylindres.

C) Fonctions réalisées :

Cette requête n'a aucune action sur le système.

8 DISPOSITIF "CHIEN DE GARDE IOCS16"

8.1 GENERALITES :

Le dispositif "CHIEN DE GARDE IOCS16" consiste à associer à chacune des FU d'une application un chien de garde limitant ainsi chaque échange à une durée maximum au delà de laquelle l'échange est abandonné.

Ce dispositif vise essentiellement la résolution des cas de blocage consécutifs aux problèmes de connexion des Coupleurs (absence de coupleur, déconnexion de bac extension etc...).

En aucun cas il ne peut se substituer au chien de garde dont sont pourvus les coupleurs. Sans être exceptionnelle son utilisation doit être limitée pour les raisons suivantes :

- précision réduite du système (environ 40 milli-secondes)
- introduction d'un overhead supplémentaire
- valeur du chien de garde fixe (génération)
- pour un périphérique fonctionnant en mode caractère, le chien de garde n'est pas réarmé à chaque transfert.

Ce dispositif est optionnel et n'affecte que les échanges dont les FU sont déclarées "surveillées".

Remarque :

S'il est utilisé, il occupe un point d'entrée, dans le driver horloge (Fonction Spéciale de Positionnement n° '43).

8.2 PRINCIPE DE FONCTIONNEMENT

8.2.1 Déclaration des FU surveillées :

S'effectue lors de la génération à l'aide de macro-instructions de GI016. A l'issue de celle-ci, trois tables sont créées :

- FUSURV : table des FU surveillées
- DOGINI : table des valeurs des chiens de garde
- DOGCOM : table des comptages

Voir description dans M.U IOCS16 § 1.3

8.2.2 Initialisation de la surveillance :

Lors du "CLEAR IOCS16" et à condition qu'une FU au moins ait été déclarée surveillée, s'effectuent :

- la communication de l'adresse de la séquence "déclenchement TIME-OUT" au driver horloge temps réel
- l'initialisation du sablier en fonction du réglage de l'horloge temps réel.



Lors de l'initialisation d'un échange sur une FU déclarée surveillée, le chien de garde est armé.

Si la fin d'échange n'intervient pas avant la durée fixée, IOCS16 tue l'échange concerné fournissant ainsi un compte-rendu d'erreur à l'utilisateur.

