

• SPS 5

• SYSTEME D'EXPLOITATION

• NOYAU DE TEST

Matériel

SUJET : Ce manuel décrit la mise en oeuvre, l'utilisation  
et les objectifs du système de test sur Solar.

OBSERVATION

VERSION LOGICIEL

DATE : NOVEMBRE 1987

MANUEL D'UTILISATION

- SPS 5
  
- SYSTEME D'EXPLOITATION •
  
- NOYAU DE TEST •

-----

Matériel •

SUJET : Ce manuel décrit la mise en oeuvre, l'utilisation •  
et les objectifs du système de test sur Solar.

OBSERVATION : •

VERSION LOGICIEL : •

DATE : NOVEMBRE 1987

(C) Bull 1986  
Dépot légal  
1er trimestre 1986

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 à la fin du présent manuel.

Ce document est fourni à titre d'information seulement. Il n'engage pas la responsabilité de Bull 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	AVERTISSEMENT .....	1.1
2	BUT DU SYSTEME DE TEST SPS 5 .....	2.1
2.1	TEST DE BON FONCTIONNEMENT/AIDE AU DEPANNAGE .....	2.1
2.2	MISE EN OEUVRE COMMUNE .....	2.2
2.3	SUPPORT DES PROGRAMMES DE TEST .....	2.2
2.4	PROGRESSIVITÉ DES TESTS .....	2.2
2.4.1	Au niveau de l'ensemble des programmes de test : notion de Noyau.....	2.2
2.4.2	Enchaînement logique des programmes de test .....	2.3
2.5	DIFFÉRENTES VERSIONS DU NOYAU .....	2.5
3	GENERALITES SUR LE NOYAU DE TEST .....	3.1
4	UTILISATION DES PROGRAMMES DE TEST .....	4.1
4.1	MISE EN OEUVRE DES PROGRAMMES DE TEST .....	4.1
4.1.1	Cas du disque à cartouche .....	4.1
4.1.2	Cas du disque souple 5"1/4 .....	4.2
4.1.3	Cas du disque souple 8" .....	4.2
4.1.4	Cas du disque à cartouche sur SPS 5 bi-processeur .....	4.4
4.1.5	Cas de la cassette .....	4.5
4.1.6	Cas des bandes magnétiques .....	4.6
4.2	STRUCTURE D'UN PROGRAMME DE TEST .....	4.6
4.2.1	Conversationnel de test .....	4.6
4.2.2	test .....	4.7
4.2.3	Phase de test .....	4.7
4.3	REPLISSAGE DE LA PHASE DE TEST .....	4.8
4.3.1	Clés .....	4.8
4.3.2	Passage des clés.....	4.9
4.4	CLÉS STANDARD .....	4.11
4.4.1	Recette REC.....	4.12
4.4.2	Recette RNS.....	4.14
4.4.3	BRL n x (Branch Line).....	4.15
4.4.4	PSW programme status word .....	4.15
4.4.5	STO - RST Store - Restore .....	4.16
4.5	L'APPEL OPÉRATEUR (BREAK) .....	4.17
4.6	MESSAGES D'ERREUR .....	4.17

4.6.1	Classes de Messages d'erreur	4.17
4.6.2	Libellé des messages d'erreur	4.17
4.7	SURVEILLANCE PERMANENTE DES INTERRUPTIONS	4.18
5	ANNEXE : CHARGEMENT DES PROGRAMMES DEPUIS UNE BANDE MAGNÉTIQUE	5.1
5.1	BUT	5.1
5.2	MOYENS NECESSAIRES	5.1
5.2.1	Matériel	5.1
5.2.2	Logiciel	5.1
5.2.3	Documentation	5.1
5.3	MISE EN OEUVRE	5.2
5.3.1	Chargement et Lancement	5.2
5.3.2	Clé CONF - Configuration	5.2
5.3.3	Clé REWI - Rebobinage	5.4
5.3.4	Clé TEST - Lancement d'un Test	5.5
5.3.5	Clé PROG - Lancement d'un Programme	5.6
5.3.6	Exemple d'utilisation	5.7
5.3.7	Erreurs détectées	5.7
5.3.8	Changement de l'UC de traitement du SPS 5 bi-processeur	5.8
5.3.9	Implantation de TESTEM	5.9
5.3.10	Contraintes d'implantation des Tests	5.10
5.3.11	Contraintes d'implantation des Programmes	5.10
5.3.12	Liste de la bande magnétique	5.11
5.3.13	Adjonction d'un test ou d'un programme sur le bande	5.11

## 1 AVERTISSEMENT

Les programmes de test des calculateurs GAMME SPS 5 fonctionnent sous le contrôle du noyau de test SPS 5. Par construction, la mise en oeuvre de n'importe lequel des tests de la série SPS 5 est identique. Cette mise en oeuvre, les buts recherchés par ce système de test et les différentes possibilités d'utilisation de ce système sont explicités dans le présent manuel. Elles seront supposés connues du lecteur d'un manuel d'utilisation d'un programme de test.

On ne peut utiliser des programmes de test de la série SPS 5 sans avoir pris connaissance du présent manuel.

## 2 BUT DU SYSTÈME DE TEST SPS 5

Les buts assignés au système de test de la série SPS 5 sont les suivants

Les programmes de test doivent

- Permettre à la fois le test de bon fonctionnement et le dépannage du module matériel testé,
- Avoir une mise en œuvre commune,
- Être disponible sur plusieurs types de support,
- Utiliser la structure modulaire de la série SPS 5,
- Être progressifs,
- Pouvoir s'exécuter en demandant le moins de mémoire possible.

### 2.1 TEST DE BON FONCTIONNEMENT/AIDE AU DÉPANNAGE

Les programmes de test de la série SPS 5 sont fonctionnels afin de tester le module matériel en utilisant les fonctions programmables du module telles qu'elles sont définies dans la partie "programmation" du manuel d'exploitation du module matériel testé. Dynamiques parce que le module est testé dans le même environnement matériel que celui où il est utilisé.

Les programmes permettent à la fois :

- de vérifier que le module testé répond à ses spécifications. Qu'il fonctionne correctement pour les performances annoncées et pour une utilisation normale client,
- de dépanner les modules ne fonctionnant pas correctement par des moyens matériels et logiciels prévus pour le test lors de la conception de la série SPS 5.

Pour un module matériel donné, le test correspondant fait l'un ou l'autre des travaux suivant la volonté de l'utilisateur. Dans les manuels de chaque programme de test, nous avons différencié ces deux approches en parlant de NIVEAU 1 et de NIVEAU 2.

On appelle niveau 1, la partie de bon fonctionnement d'un programme de test. Il permet d'effectuer les actions suivantes :

- Recette de bon fonctionnement,
- Vieillessement du matériel,
- Contrôle d'entrée des périphériques,
- Entretien préventif.

Il concerne tous les types d'utilisateur des Programmes de test et surtout le client qui veut, avant d'appeler l'après-vente pour le dépannage de son matériel, obtenir le maximum de précisions sur la panne afin que l'intervention de l'après-vente soit la plus efficace possible.

Le NIVEAU 2 ne doit être utilisé que par les clients "OEM" et les agents d'après-vente. Il permet la recherche et le dépannage du défaut.



Les programmes de test de la série SPS 5 respectent un interface de dialogue avec le Noyau de Test. Quel que soit le programme de test, la mise en oeuvre est commune. Seul le contenu du test varie avec le module testé. Une fois la mise en oeuvre commune connue, l'utilisateur peut facilement passer d'un programme de test à l'autre.

### 2.3 SUPPORT DES PROGRAMMES DE TEST

Les programmes de test sont disponibles sur plusieurs supports

- Cassette,
- Disque à cartouche,
- Disques souples,
- Bande magnétique.

Pour l'utilisation de ces supports, voir le paragraphe 4.1.  
Pour la bande magnétique, voir l'annexe en chapitre 5.

### 2.4 PROGRESSIVITE DES TESTS

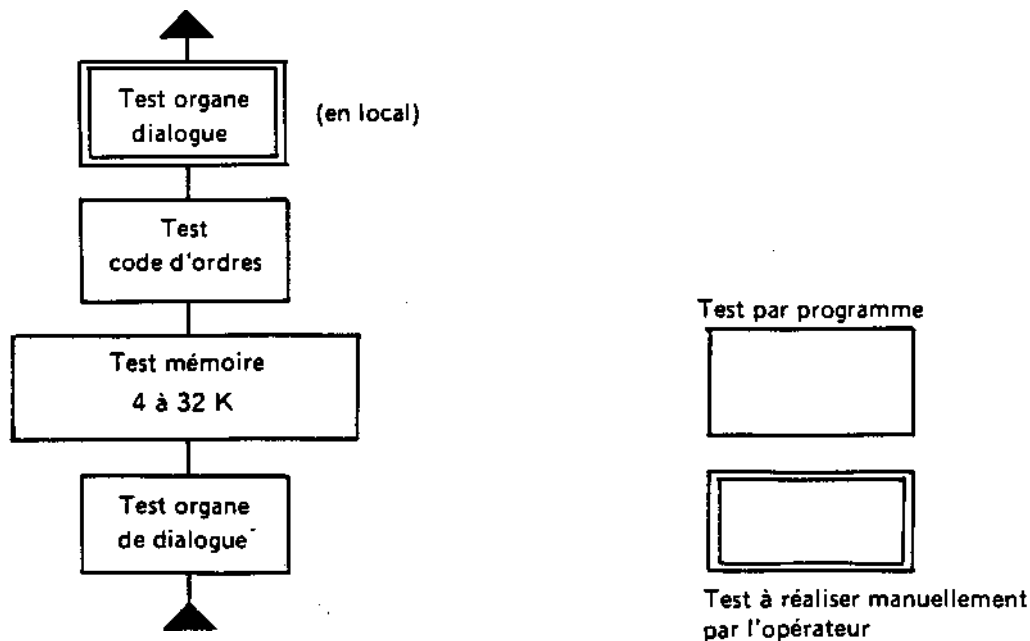
#### 2.4.1 Au niveau de l'ensemble des programmes de test : notion de Noyau Matériel

Auto-test du Noyau matériel de la machine puis test des autres modules ou de l'ensemble des modules suivant un enchainement tel, qu'il ne remet pas en cause le bon fonctionnement des modules déjà testés.

Par noyau matériel on entend un système constitué

- d'un processeur SPS 5,
- d'une mémoire centrale de capacité au moins égale à 32K,
- d'un organe de dialogue fonctionnant en programmé simple.

Auto-test du Noyau Matériel



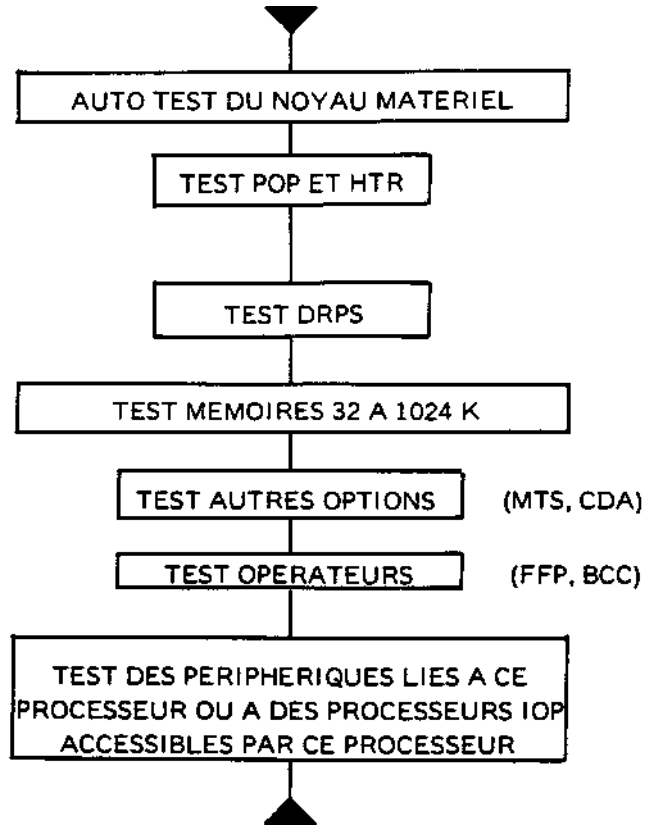


## 2.4.2 Enchaînement logique des programmes de test



Le contenu du Noyau matériel donné, on veut définir l'enchaînement logique des programmes de test.

Organigramme :





## o-test du Noyau Matériel

L'auto test du Noyau matériel s'effectue en 4 étapes.

### ETAPE 1

Validation de l'organe de dialogue.

On peut commencer par valider en local la visualisation. L'étape 2 qui utilise l'organe de dialogue permettra de savoir s'il marche correctement en programmé simple.

### ETAPE 2

Exécution des programmes de test du code d'ordre. C'est une série de 4 programmes de test.

Ces tests sont écrits avec un ensemble réduit d'instructions de base. La liste de ces instructions est la suivante

LA - STA - AD - EOR - AND - BR - BSR

JMP - JC - JV - JAE

SCLS

LR - XIMR - LRP

SST - RST - ACQ

SIO télétype - SVC - RSR - RDOE - WOE - RBP - SBP

ACK - PSR - PLR

Parmi ces instructions, 7 (XIMR, SST, RST, ACQ, ACK, PER, PLR) sont utilisées très rarement.

Les programmes de test du code d'ordre travaillant en 2 temps

- temps 1 : test des instructions de base par elles-mêmes
- temps 2 : test de toutes les autres instructions.

Si dans le programme de test des instructions de base par elles-mêmes, une instruction ne marche pas, cela ne veut pas dire forcément qu'elle est mauvaise. Mais que cette instruction ou une autre instruction ou plusieurs parmi les instructions de base ne marchent pas.

A la fin du test des instructions de base par elles-mêmes, on peut arriver à l'une des deux situations suivantes

- Elles marchent : on fait passer les autres programmes de test du code d'ordres. Si dans ces programmes une instruction ne marche pas, on est sûr du diagnostic puisque les tests du code d'ordres sont programmés uniquement avec les instructions de base.
- Elles ne marchent pas : il faut dépanner le matériel jusqu'à ce que le test des instructions marche. Ce peut être le processeur, ce peut être aussi la mémoire 7

Le Noyau spécial instructions (référence : 1 158 000 02) programmé avec les instructions de base est identique au Noyau normal, hormis le fait qu'il accepte des clés à 4 caractères.

## ETAPE 3

Validation de la mémoire de 4 à 32 K par le test approprié.

## ETAPE 4

Validation complète de l'organe de dialogue par son programme de test.

### B) Enchaînement des autres programmes de test

Test du reste des fonctions liées au processeur. Le noyau matériel complètement validé, il faut valider les autres parties du processeur et ses options. Validation tout d'abord de l'horloge temps réel (HTR) et du pupitre opérateur (POP) s'il existe.

Validation ensuite de l'option DRPS. Une fois le DRPS validé, test de la mémoire au delà de 32K.

Test ensuite de l'option SCHEDULER MTS du processeur si elle existe. L'utilisateur doit ensuite tester les opérateurs qui entourent et complètent le processeur (flottant - FFP, opérateur BCC, etc...)

Test de la périphérie : c'est-à-dire test des coupleurs, qui sont accessibles soit directement par le processeur sur lequel se déroule les programmes de test, soit à travers des processeurs (IOP 16). Pour cela 2 types de programme de test

- 1- Les programmes testant un coupleur et lui seul, sans savoir ce qu'il a derrière.  
Exemples : test de l'interface universelle, des coupleurs asynchrones.
- 2- Les programmes testant, le coupleur, le ou les périphériques qui lui sont associés ainsi que la liaison coupleur périphérique. Le périphérique pouvant être un formateur gérant lui-même plusieurs périphériques.  
Exemples
  - test du lecteur de carte : il teste le coupleur et le lecteur de cartes
  - test du disque à cartouche : il teste le coupleur et 1 à 4 unités de disques à cartouche
  - test du disque souple 8" : il teste le coupleur, le formateur et 1 à 4 unités de disques souples.

## 2.5 DIFFERENTES VERSIONS DU NOYAU

Le noyau de test SPS 5 existe en plusieurs versions

- NOYAU DE TEST : réf. 1 158 000 01  
Version de base pour tous les tests autres que les tests des instructions
- NOYAU DE TEST INSTRUCTIONS : réf. 1 158 000 02  
Version de base pour les tests des instructions
- NOYAU DE TEST bi-processeur 1.158.000.06  
Version de base pour tous les tests autre que les tests des instructions.
- NOYAU DE TEST instructions bi-processeur 1 158 000 06  
Version de base pour les tests des instructions.

## 3 GENERALITES SUR LE NOYAU DE TEST

Le noyau de test SPS 5, programme de taille 1,25 K mots, est un moniteur de test, autonome, conversationnel, réentrant.

Aspect moniteur

C'est un moniteur auquel les programmes de test demandent des services. Il comprend donc des sous-programmes d'intérêt commun que l'on peut appeler à partir de tout programme de test à condition que l'on respecte l'interface noyau <---> programme de test défini dans "le manuel de programmation des programmes de test sous noyau".

Ces sous-programmes sont des

- sous-programmes de gestion et d'enchaînement
  
- sous-programmes nécessaires à tout programme de test (conversion hexadécimale ou décimale --> binaire, générateur de nombres aléatoires, générateur de nombres pseudo aléatoires etc...)
  
- sous-programmes de gestion de l'organe de dialogue.

Le noyau offre un haut degré de protection du programme de test vis-à-vis des fausses manœuvres de l'opérateur.

Aspect autonome

Le noyau ne suppose pas, pour fonctionner, l'existence préalable d'un système d'exploitation, mais il peut s'insérer dans le cadre d'un tel système (par exemple : cas où les programmes de test ont comme support le disque à cartouche). Il gère les mémoires débanalisées et les 16 PST hardware.

Aspect conversationnel

L'activation et l'exploitation normale d'un programme de test se font avec un organe de dialogue. Tout programme de test est découpé en "clés". Toute clé est activable de façon conversationnelle. On peut activer soit une clé, soit plusieurs clés, soit des ensembles de clés figés lors de la conception du test. On peut donc de façon conversationnelle se construire de véritables programmes.

L'organe de dialogue est en principe le téléimprimeur de service, ou plus exactement le clavier (TK) et la feuille (TS) de ce téléimprimeur. Mais l'organe de dialogue peut être aussi une combinaison coupleur + périphérique dont la programmation est compatible avec celle du téléimprimeur de service et de même adresse '17F8.

Exemple

ASVO1 et (visualisation ou téléimprimeur)

MXPO4 et (visualisation ou téléimprimeur)

L'organe de dialogue est gérée en programmé simple pour ne pas avoir d'autres causes d'interruptions que celles du ou des modules testés. Si l'organe est un téléimprimeur de service à mise en route automatique, le noyau gère sa mise en route lorsqu'il veut éditer des messages.

## 4 UTILISATION DES PROGRAMMES DE TEST

### 4.1 MISE EN OEUVRE DES PROGRAMMES DE TEST

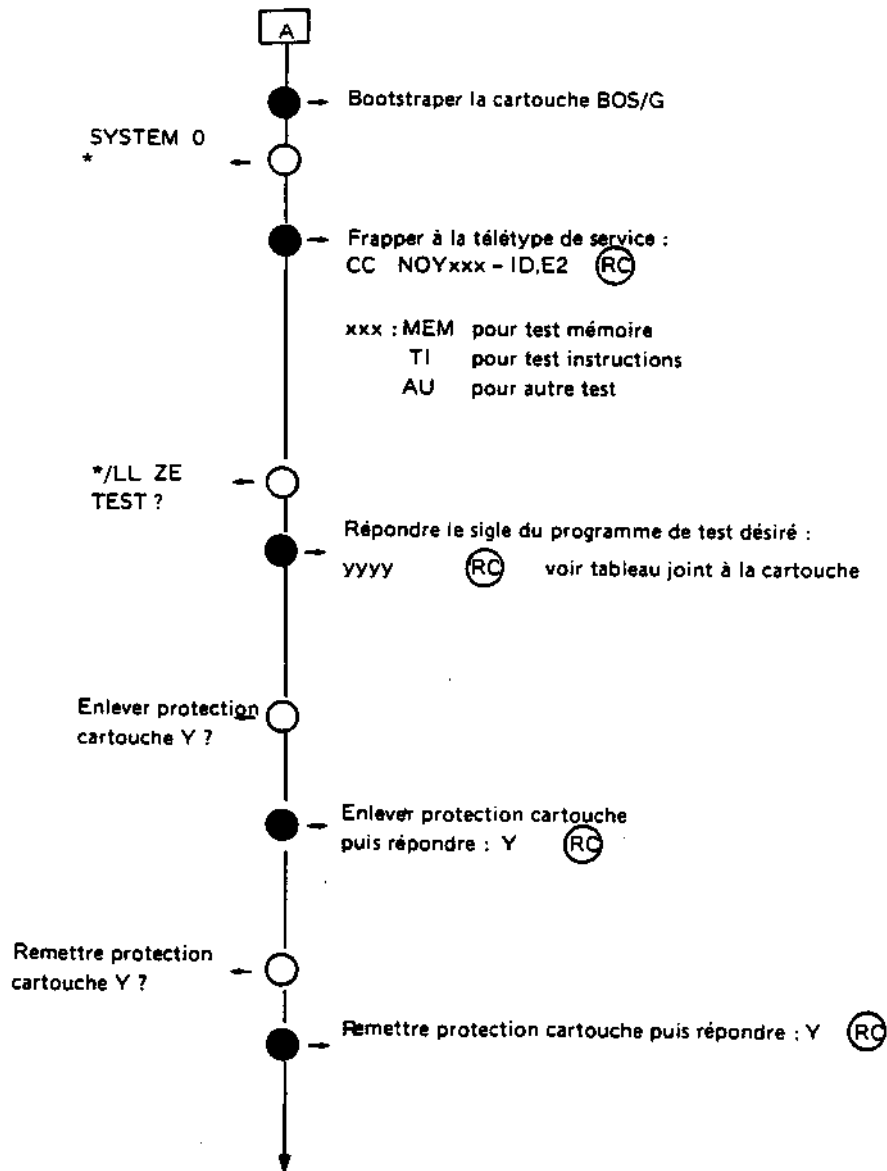
La mise en oeuvre des programmes de test varie suivant le type de leur support.

#### 4.1.1 Cas du disque à cartouche

Tous les programmes de test sont sous forme d'image mémoire

Réponse de la télétype

Manipulation opérateur



Puis on enchaîne avec le dialogue spécifique au programme test. Pour abandonner ce programme de test et lancer un autre programme : Revenir en **A**

Les programmes de tests sont stockés sur :



Cartouche	BOS/G
Unité fonctionnelle	E2
Catalogue	I0
Fichier	Image mémoire

Ils s'implantent en :

Noyau test mémoire : '1000  
Programme test mémoire : '1508

Noyau test instructions : '170  
Programme test instruction : '8D0

Noyau normal : '170  
Programme de test : '680

En cas d'erreur ou de défaut, apparaît un message du type :

ERREUR RÉPONSE ou DÉFAUT DISQUE

Dans ces cas on reformulera la réponse à la question TEST ?, ou l'on repartira en A.

#### 4.1.2 Cas du disque souple 5"1/4

Ce cas est identique à celui du disque à cartouche, à cela près que les tests sont répartis sur plusieurs disques souples. Il faudra donc sélectionner le disque souple contenant le test que l'on veut exécuter, le bootstrapper. On obtient le message suivant sur l'organe de dialogue :

FM-DM-KD-KM-MK ?

Répondre : FM (RC)

On obtient le message :

SYSTEM 0

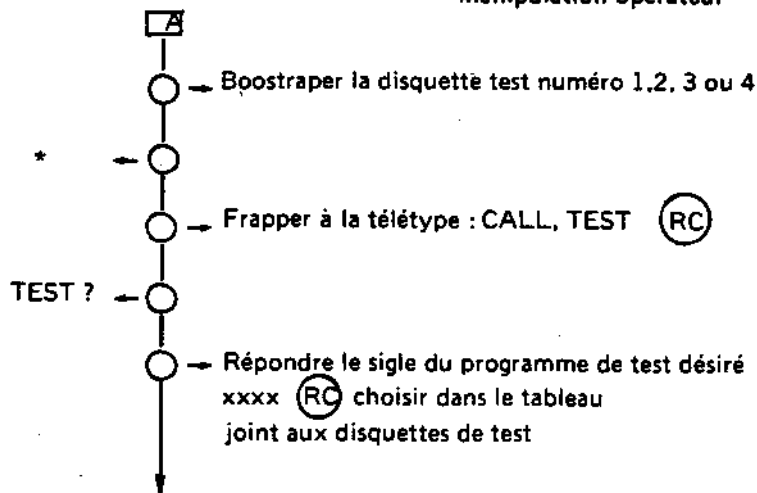
\*

Pour le reste, se reporter aux paragraphes traitant du disque à cartouche.

#### 4.1.3 Cas du disque souple 8"

Réponse de la télétype

Manipulation opérateur



Puis on enchaîne avec le dialogue spécifique au programme de test. Pour abandonner ce programme de test et lancer un autre programme

Revenir en A

(En cas d'erreur de réponse ou de défaut d'échange apparaîtra le message "DEFAUT DISQUE" ou "ERREUR REponse")

Exemple

```
* CALL, TEST
TEST ? MEM8
REF 1.158.232
```

|
|
|
|
|

On fait : STOP, INI, LORD, RUN au pupitre

```
*CALL, TEST
TEST?T11
REF 1 158 200
```

|
|
|
|
|

Les programmes :

Disquettes contenant	BOS/C, FDM
	FDUP 1
	FDUP 2

Fichiers programmes de test : image mémoire

Adresses d'implantation :	Noyau Test mémoire	:	'1000
	Programme test mémoire	:	'1508
	Noyau test instructions	:	'170
	Programme test instructions .	:	'8D0
	Noyau normal	:	'170
	Programme de test	.	'680

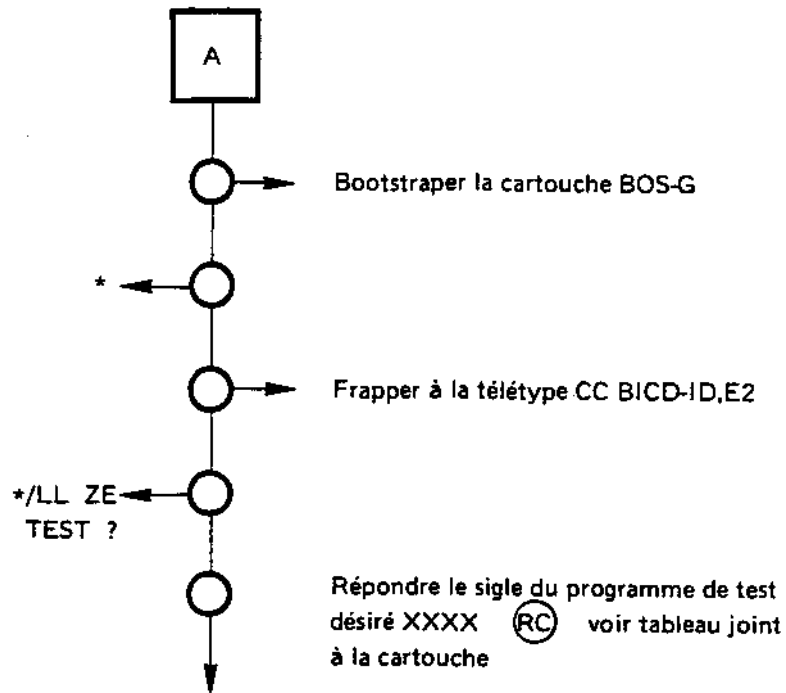
4.1.4 Cas du disque à cartouche sur SPS 5 bi-processeur



Le chargement des tests dans la mémoire de l'UC de traitement est fait par le frontal.

Réponse de la télétype

Manipulation opérateur

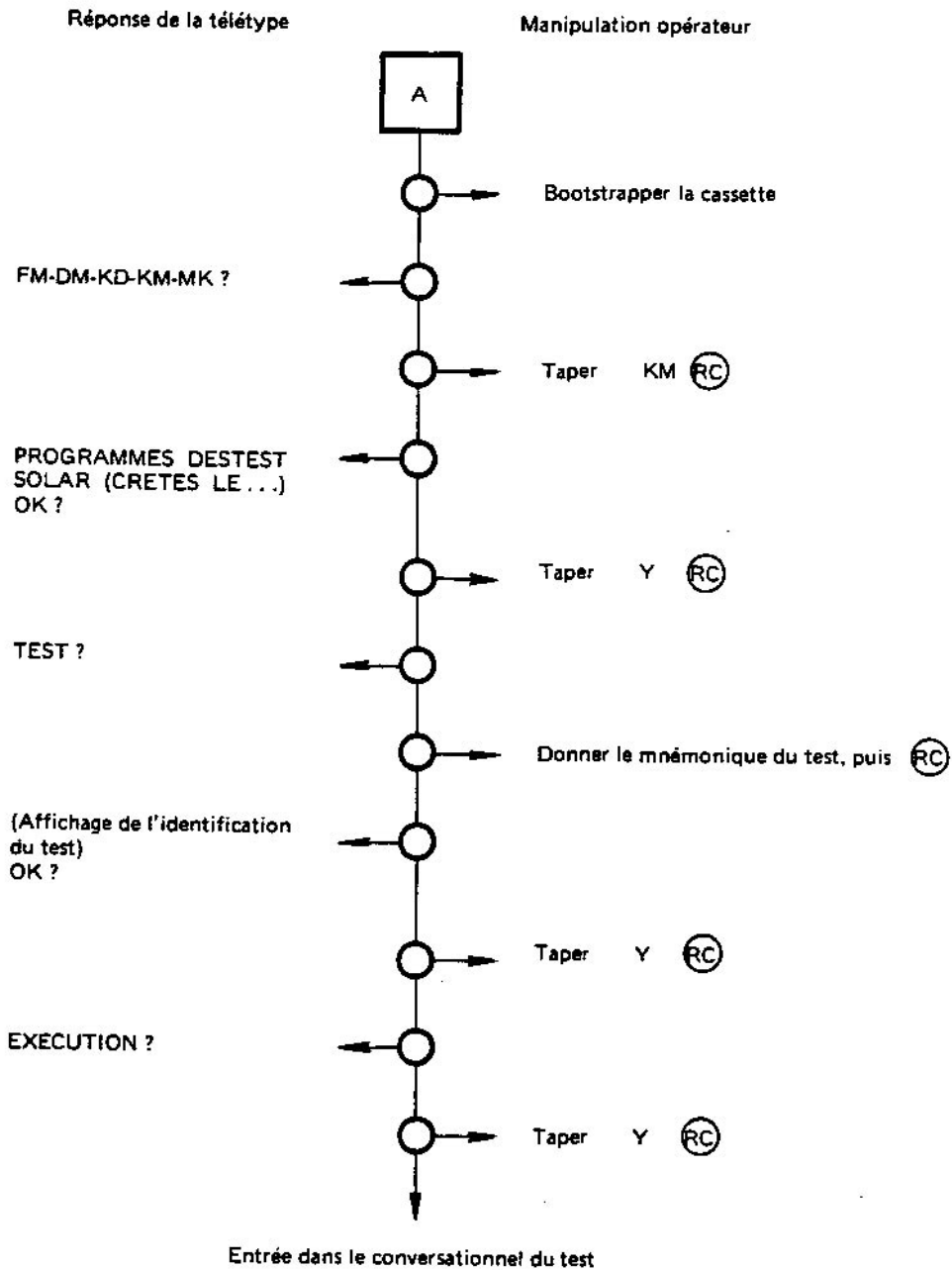


Puis on enchaîne avec le dialogue spécifique au programme de test. Pour abandonner ce programme et en lancer un autre : revenir en A.

Les adresses d'implantation dans la mémoire de l'UC de traitement sont identiques à celles du cas standard.

En cas d'erreur ou de défaut apparaît un message du type ERREUR REPONSE ou DÉFAUT DISQUE. Dans ces 2 cas on reformule la question TEST ?





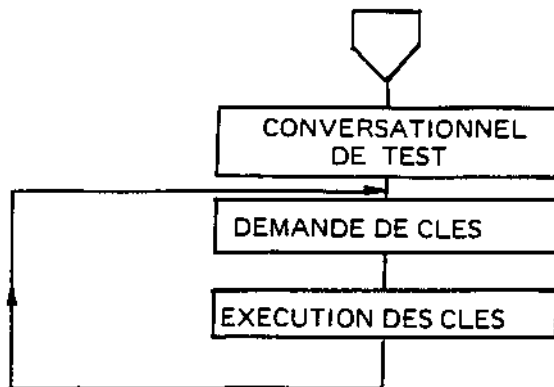
En cas de défaut, ou pour changer de test, revenir en A.  
Les fichiers inscrits sur la cassette de test regroupent le programme de test proprement dit et le noyau qui lui correspond.  
Ni l'opérateur, ni le micro-programme de chargement n'a donc à se préoccuper de la version du noyau. L'adresse d'implantation de l'ensemble Noyau-programme est '400, et l'adresse de lancement figure dans le descripteur du fichier.



Le cas particulier du chargement des programmes à partir d'une bande magnétique 800 ou 1600 bpi est traité dans le dernier chapitre.

#### 4.2 STRUCTURE D'UN PROGRAMME DE TEST

Tout programme de test est structuré de la manière suivante



Un conversationnel de test demande à l'opérateur de préciser tous les renseignements nécessaires pour le lancement du "test"

Le programme demande ensuite à l'opérateur le travail qu'il veut faire sous la forme d'une liste de clés (phase de test). Celle-ci commence lorsque le programme demande DONNEZ VOS CLÉS.

Une fois cette demande terminée, on lance l'exécution du travail demandé. Cette exécution terminée, le programme demande à nouveau à l'opérateur le travail qu'il veut faire.

##### 4.2.1 Conversationnel de test

C'est un ensemble de questions posées à l'opérateur auquel il doit répondre. Les réponses sollicitées peuvent être

- soit Y (yes) et N (no) et toute autre réponse force le programme à reposer la question.

Exemple LDC ? U "RC"

LDC ? Y "RC"

- soit un chiffre qu'il peut indifféremment exprimer en hexadécimal ou en décimal signé ou non

Le programme repose la question

- s'il ne peut convertir ce chiffre
- s'il a fait un contrôle négatif entre le chiffre donné par l'opérateur et les bornes assignées par le programmeur
- si le chiffre donné par l'opérateur n'est pas cohérent avec d'autres réponses de l'opérateur.

Si un autre type de réponse est sollicité, la question indique dans son libellé les réponses possibles.

Exemple

TYPE IMPRIMANTE (L = MODULE 1.159.330 - D = MODULES 1.159.331,332,333) ?

Ici on doit répondre L ou D, sinon le programme repose la question. Les contrôles de cohérence de paramètres peuvent amener le programme à reposer tout ou partie des questions. Tout manuel d'utilisation d'un programme de test possède un organigramme détaillé du conversationnel de test où sont expliqués les contrôles effectués sur les réponses de l'opérateur.

#### 4.2.2 Test

Un test représente l'ensemble des actions entreprises sur un module matériel testé et sous un conversationnel donné :

Si l'opérateur :

- pendant la définition d'un test
- pendant la définition d'une phase de test
- en cours d'exécution de la phase de test

décide de redéfinir les conditions du test, c'est-à-dire son conversationnel, il répond "\$" à la première question qui lui est posée quelle que soit cette question.

En résumé :

- un test commence en entrant pour la première fois dans un programme de test, ou après avoir répondu "\$" à n'importe laquelle des questions posées dans le programme de test.
- le test se termine en répondant "\$" à n'importe laquelle des questions posées dans le programme de test.

#### 4.2.3 Phase de test

La définition d'une phase de test ne se fait que lorsque le test a été défini. Une phase de test est une certaine quantité de travail demandée par l'opérateur sous forme de clés. L'opérateur peut demander n'importe quelle clé existant dans le programme dans n'importe quel ordre, sauf pour les clés standard REC et RNS qui doivent être utilisées seules.

En résumé, une phase de test commence par DONNEZ VOS CLÉS et on y accède dans les cas suivants

- après la définition d'un test
- quand la phase de test précédente est terminée.

Une phase de test se termine :

- Après l'exécution de la dernière clé de la phase de test
- Sur un appel organe de dialogue

## 4.3 REMPLISSAGE DE LA PHASE DE TEST

### 4.3.1 Clés

Tout programme de test comporte un ensemble de clés. Une clé est un sous-programme dont l'exécution peut être lancée par l'organe de dialogue du calculateur. Une clé est identifiée par son mnémonique de 3 caractères ASCII (noyau spécial instructions : de 2 à 4 caractères ASCII).

Il y a deux types de clés au niveau 1

- les clés standards (S). Ce sont des clés communes à tout programme de test. Leur mnémonique est alphabétique  
Exemple : REC, RUS
- les clés action (A). Ce sont des clés spécifiques à chaque programme de test. Leur mnémonique est numérique.  
Exemple : 101, 102

Lorsque le concepteur prépare le test d'un module matériel, il fait la liste de toutes les actions à entreprendre sur ce module dans le cadre de sa programmation «normale» pour mettre en évidence son bon fonctionnement, ainsi que la liste des actions «anormales» possibles pour montrer qu'il détecte bien et/ou récupère bien les défauts. Cette liste va dans un ordre logique du plus simple au plus compliqué.

Le découpage de cette liste en modules logiques donne la liste des clés action.

Lors de l'exécution d'une clé action,

- s'il n'y a pas eu d'erreur au cours de l'exécution de la clé action XXX, le message suivant est imprimé : FIN XXX OK (fin correcte)
- s'il y a eu au moins une erreur en cours d'exécution de la clé action xxx, le message suivant est imprimé : FIN XXX NOK (fin incorrecte =non OK)

Pour chaque programme, la liste des clés actions disponibles au niveau 1, leurs caractéristiques, le nombre de paramètres de chaque clé, son temps d'exécution, sont résumés par un tableau dans le manuel d'utilisation, et/ou d'exploitation.

Une clé peut avoir zéro, un ou plusieurs paramètres (mais inférieur à 10). Ces paramètres sont numériques. Ce peut être indifféremment des nombres décimaux signés, des nombres décimaux non signés, des nombres hexadécimaux (c'est-à-dire le caractère quote '"' suivi de 4 chiffres hexadécimaux 0 ... 9A ... F)

Le noyau se charge de la reconnaissance du type du paramètre et de sa conversion.  
Exemple

2 '2 +2 '02 02 +02 sont équivalents au nombre décimal 2 mais ' 2 n'est pas considéré comme un paramètre correct.

#### 4.3.2 Passage des clés



Dans tous les exemples qui vont suivre, on suppose être dans un programme de test possédant les clés suivantes :

- 101        0        paramètre
- 102        0        paramètre
- LAD        1        paramètre

. On ne peut commencer à définir une phase de test que lorsque le programme demande DONNEZ VOS CLÉS et affecte un numéro de ligne 01.

```
DONNEZ VOS CLÉS
01
```

L'opérateur donne alors la première clé qu'il veut faire exécuter suivie des paramètres de la clé. Exemple : demande de la clé 101 sans paramètre

```
DONNEZ VOS CLÉS
01 101 "RC"
```

Après le retour chariot "RC" le noyau contrôle l'existence de cette clé dans le programme de test, la coïncidence entre le nombre de paramètres déclaré de la clé et le nombre de paramètres donné par l'opérateur, la possibilité de faire la conversion de ces paramètres en binaire. Si la clé est acceptée le programme enregistre cette clé dans une table, fait progresser de 1 le numéro de ligne et attend une nouvelle clé. Exemple : 101 a été acceptée par le programme et le programme attend une deuxième clé.

```
DONNEZ VOS CLÉS
01 101 "RC"
02 <----- nouveau numéro de ligne
```

Par contre si la clé n'est pas acceptée elle n'est pas enregistrée, et le numéro de ligne ne progresse pas. Donc s'il y a plusieurs lignes de clés ayant le même numéro, c'est la dernière ligne qui est enregistrée dans la phase de test et elle seule.

Exemple : les 5 phases de test I, II, III, IV, V sont équivalentes.

```
I     DONNEZ VOS CLÉS
      01 LAD 10 "RC"
      02
```

```
II     DONNEZ VOS CLÉS
      01 LAD RC                   cause du refus : pas
                                  de paramètre
      01 LAD 10 "RC"
      02
```

```
III    DONNEZ VOS CLÉS
       01 LAD 25 'AE "RC"        cause du refus : trop
                                  de paramètres
       01 LAD 10 "RC"
       02
```

```
IV     DONNEZ VOS CLÉS
       01 LAD 'AG "RC"           cause du refus :
                                  conversion du paramètre
                                  impossible
       01 LAD 'A "RC"
       02
```

```
V   DONNEZ VOS CLÉS
    01 LAE RC                Cause du refus : la
                                clé LAE n'existe pas
                                dans le programme de test
    01 LAD +010 "RC"
    02
```

. Le buffer d'entrée de toutes les réponses de l'opérateur est unique et a une longueur donnée pour chaque programme. Si on tente de dépasser les limites de ce buffer, la réponse est refusée et la question reposée.

Exemple : La longueur du buffet d'entrée est de 10 caractères.

```
DONNEZ VOS CLÉS
01 AAAAAAAAAA                <----- Refus
01
```

. Quand l'opérateur décide de clore la définition de la phase de test, après qu'il ait passé la dernière clé et lorsque le programme lui demande une nouvelle clé, il répond : "RC" exemple :

```
DONNEZ VOS CLES
01 101 "RC"
02 102 "RC"
03 RC
```

Le programme a enregistré comme phase de test les clés 101 et 102. Le programme lance leur exécution sous le contrôle des paramètres de la clé (S'il y en a) et des réponses fournies par l'opérateur lors du conversationnel de test. Lorsque la phase de test est finie, le programme demande la définition d'une nouvelle phase de test.

```
DONNEZ VOS CLES
01 101 "RC"
02 102 "RC"
03 RC                phase de test 1
FIN 101 OK
FIN 102 OK
DONNEZ VOS CLES    phase de test 2
01
```

. la table enregistre une phase de test a une longueur donnée pour chaque programme de test. Cette longueur est indiquée dans le manuel d'utilisation de chaque programme de test. Si l'opérateur dépasse le nombre maximal de (clés + paramètres) que l'on peut enregistrer, le noyau abandonne la définition de la phase de test et demande la définition d'une nouvelle phase de test.

Exemple : le programme accepte un nombre maximal de (clés + paramètres) égal à 3

```
DONNEZ VOS CLES
01 LAD 10
02 102
03 102 <-----          abandon sans exécution
DONNEZ VOS CLES
01
```

En fait cela arrive peu souvent, les tables étant largement dimensionnées en fonction du nombre total de clés du programme de test.

. Pour corriger une erreur de frappe ou pour retourner en début : de test ou de phase de test, on dispose des caractères suivants

"↑" suppression du damier caractère frappé

"←" suppression de la ligne frappée

"#" retour à DONNEZ VOS CLES

"\$" retour au conversationnel du test

Exemple

```

DONNEZ VOS CLÉS
01 10 A11 "RC"
02 RC
FIN 101 OK

DONNEZ VOS CLÉS
01 A02 ← 102 "RC"
02 "RC"
FIN 102 OK

DONNEZ VOS CLÉS
01 103 "RC"
# 02 104 "RC"
03 1 #
DONNEZ VOS CLÉS

DONNEZ VOS CLÉS
$ 01 10 $
CAPACITÉ MÉMOIRE ?

```

#### 4.4 CLÉS STANDARD

Elles sont au nombre de 6 et sont communes à tous les programmes de test.

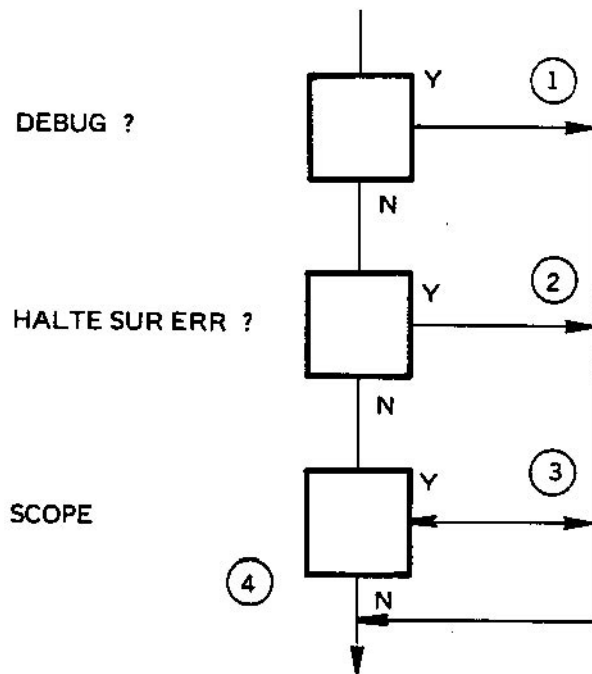
Mnémorique	Nombre de paramètres		But de la clé
REC	0	Recette	Recette de bon fonctionnement
RNS x	1	Recette non	Recette longue durée sans opérateur
		Stop	
BEL n x	2	Branch line	Branchement à la ligne numéro n x fois
PSW	0	PGM Status	Impose un mode de fonctionnement
		Word	
STO	0	Store	Stocke la phase de test décrite
RST	0	Restore	Restitue la phase de test décrite lors de l'utilisation de la clé STO

4. Bull Le REC

C'est une recette de bon fonctionnement nécessitant la présence d'un opérateur. Après le découpage du programme de test en clés action, il est figé dans une table, la liste des **clés action nécessaires** à la recette de bon fonctionnement (table de la recette REC). Cette liste présente un ordre logique, c'est-à-dire allant des tests élémentaires aux tests sollicitant le plus de matériel testé. REC contient un ensemble de clés action. Pour connaître le contenu de la recette REC de chaque programme de test, il faut consulter le manuel d'utilisation du programme de test.

Lors de l'utilisation de REC, si les premières clés de la recette ne fonctionnent pas, il n'est pas forcément utile de laisser les clés se dérouler entièrement; il vaut peut-être mieux comprendre et analyser ce qui se passe. Réciproquement il n'est peut-être pas nécessaire de se limiter aux symptômes de mauvais fonctionnement des premières clés. On risque de ne pas avoir une vue d'ensemble de problème.

Lorsqu'on demande l'exécution de REC, le programme pose 3 questions à l'opérateur. Suivant les réponses, la recette REC fonctionne dans 4 modes différents.





#### A) Mode normal "4"

L'opérateur a répondu N (non) aux 3 questions. La recette s'exécute avec émission de messages d'erreur au niveau demandé par le conversationnel de test.

#### S) Mode Debug "1"

Le mode debug exécute les clés de REC testant les fonctions élémentaires du module et dont le temps d'exécution n'est pas trop long. Elles sont en général groupées en tête de la recette. La notion de durée (pas trop longue) est fonction de la complexité du module teste.

Le test par le mode debug n'est pas exhaustif. Il se donne pour but :

- Faciliter la maintenance préventive
- S'il est reconnu que le module est défectueux, de fixer le degré de gravité de la panne et de faciliter le dialogue avec l'après-vente afin de déterminer le profil des actions d'après-vente nécessaires au dépannage.

#### C) Halte sur Erreur "2"

On peut travailler en mode Halt sur erreur

- Pendant la recette si on le demande (question HALTE SUR ERRE. ? . Y)
- Pour toute les clés ou il existe un mode Halte sur Erreur implanté (voir le détail des clés actions dans la notice de fonctionnement NIVEAU 2 du test concerné). S'il n'existe pas, le fonctionnement de la clé ne sera pas altéré. Il faut demander préalablement le mode Halte sur Erreur au moyen de la clé standard "PSW".

Le mode Halte sur Erreur permet lors de l'exécution d'un clé, à la détection d'une erreur de :

- Continuer en séquence la phase de test définie jusqu'à la prochaine erreur si l'opérateur répond "Y" à la question "CONTINUER ?"
- De définir une nouvelle phase de test si l'opérateur répond "N" à cette question.

#### D) Mode Scope "3"

On peut travailler en mode scope

- Pendant la recette REC si on le demande (question SCOPE ? Y)
- Pour toutes les clés ou il existe un mode scope implanté. S'il n'en existe pas, le fonctionnement de la clé ne sera pas altéré. Il faut demander préalablement le mode scope au moyen de la clé PSW.

Le mode Scope permet lors de l'exécution d'une clé, à la détection d'une erreur (et à condition qu'il ait été implanté à cet endroit) de se reboucler sur l'échange défectueux tant que l'erreur subsiste.

Après l'émission du message d'erreur, on demande à l'opérateur s'il veut rentrer dans la boucle scope "SCOPE ?". Si l'opérateur répond "N" c'est qu'il veut boucler non pas à cet endroit mais plus loin (cas de plusieurs modes scope implantés dans une clé). Le programme continue le déroulement de la clé.

Si l'opérateur répond "Y" c'est qu'il veut boucler sur l'échange défectueux. A partir de ce moment et tant que l'on boucle les messages d'erreur sont supprimés.

On peut sortir de cette boucle.

- Par un appel sur l'organe de dialogue
  
- Parce que l'erreur a disparu.
  - A/ elle a été dépannée
  - B/ elle est aléatoire.

Dans ce cas l'opérateur est prévenu que l'on sort de la boucle par la question "CONTINUER ?"

- S'il répond "Y" le programme boucle indéfiniment même si l'erreur est aléatoire. Il ne pourra sortir de cette boucle que par un appel sur l'organe de dialogue.
  
- S'il reprend "N" le programme reprend le travail demandé au passage des clés.

Contraintes d'utilisation de REC

REC ne peut être utilisée que comme seule et unique clé d'une phase de test.

#### 4.4.2 Recette RNS

Cette recette de longue durée ne nécessite pas la présence de l'opérateur après son lancement, sauf pour le test de quelques modules fonctionnant avec l'aide d'un opérateur (exemple : lecteur/perforateur de ruban).

Cette recette est destinée aux recettes de longue durée et au vieillissement du matériel. Pour chaque programme de test il est figé dans une (table de la recette RNS) l'ensemble des clés actions nécessaires. La recette RNS est de durée infinie, cela veut dire qu'elle se reboucle indéfiniment sur elle-même.

RNS est une clé possédant un paramètre : le nombre maximum de lignes de messages dont on autorise l'édition.

```
DONNEZ VOS CLES
01 RNS 9 "RC"

02 "RC"
```

Si au cours de la recette RNS, il y a émission de messages d'erreur, RNS éditera 9 lignes de messages d'erreur, puis on continuera sans éditer de message autres que ceux de fin de clé où s'est produit un défaut.

FIN xxx NOK TOUR : 'YYYY (xxx : mnémonique de la clé; YYYY : numéro de la boucle en hexadécimal où s'est produite l'erreur).

RNS est de durée infinie et peut être arrêtée soit en rebootstrappant, soit par appel "Break" sur l'organe de dialogue. Si la recette RNS est arrêtée par appel "Break" l'organe de dialogue imprime

- le nombre total de lignes de messages d'erreur édités ou non,  
RNS (nombre limite à 'FFFF)
  
- le nombre de tours faits dans la recette RNS (nombre limité à  
FFFF)

```
RNS : NB ERR 'xxxx TOURS 'xxxx
```

Contraintes d'utilisation de RNS :

RNS ne peut être utilisé que comme seule et unique clé d'une phase clé test.

Exemple

```
DONNEZ VOS CLES
01 RNS 2 "RC"
02 "RC"
ERR (101/01) message 1
FIN 101 NOK TOUR : '0001
ERR (102/03) message 2
FIN 102 NOK TOUR : '0002
FIN 303 NOK TOUR : '0010
```

```
"Break" RNS : NB ERR '0003 TOURS '0015
DONNEZ VOS CLES
01
```

#### 4.4.3 BRL n x (Branch Line)

La clé BRL possède deux paramètres (n et x). C'est une clé qui permet de sauter à la ligne de clé de numéro (n) (x) fois. X fois peut être infini si au lieu de donner comme paramètre un nombre on donne le caractère "\*"

Exemple d'utilisation de la clé

```
DONNEZ VOS CLES
01 WRI
02 BRL 1 5
03
```

La clé WRI sera exécutée 6 fois.

```
DONNEZ VOS CLES
01 WRI
02 BRL 1 *
03
```

La clé WRI sera exécutée en permanence. On ne peut sortir de cette boucle que par un appel sur l'organe de dialogue.

La clé BRL peut être utilisée plusieurs fois dans une phase de test mais il faut que le numéro de ligne (n) donné par la clé soit inférieur au numéro de ligne de la clé.

Exemple

```
DONNEZ VOS CLES
01 101
02 BRL 1 5
03 102
04 BRL 3 2
05 BRL 1 *
06
```

Il sera exécuté en permanence la clé "101" 6 fois puis la clé "102" 3 fois.

#### 4.4.4 PSW programme status word

Cette clé permet d'imposer un état de fonctionnement pour les clés outils et actions qui la suivent. Elle permet de travailler au choix

- Avec suppression des messages d'erreur
- En mode pas à pas
- En mode Scope
- En mode Halte sur Erreur

L'état de fonctionnement donné par la clé PSW ne peut être modifié que par :

- L'exécution d'une nouvelle clé PSW
- Une nouvelle définition de phase de test

Exemple

```
DONNEZ VOS CLÉS
01 PSW RC
02 101 RC
03 RC
SUPPRES MESS ERREUR ? N
PAS A PAS ? N
SUPPRESS MESS FIN DE CLE ? N
SCOPE ? N
HALTE SUR ERR ? N
FIN 101 OK
DONNEZ VOS CLÉS
01
```

La réponse "y" à la question "SUPPRESS MESS ERREUR ?" supprime l'édition de messages d'erreur mais n'empêche pas l'édition de "FIN clé OK ou NOK".

Le mode pas à pas :  
avant l'exécution de chaque clé, le mode pas à pas demande à l'opérateur s'il veut lancer le test par la question "CONTINUER ?". Si l'opérateur répond "y" la clé sera activée. La question lui sera posée s'il répond "N".

Exemple :

```
DONNEZ VOS CLES
01 PSW
02 101
03 102
04
SUPPRES MESS ERREUR ? N
PAS A PAS ? Y
CONTINUER ? Y
FIN 101 OK
CONTINUER ? N
CONTINUER ? Y
FIN 102 OK
CONTINUER ? Y
DONNEZ VOS CLES
01
```

Le mode scope et halte sur erreur :

voir la description de ces modes dans le chapitre concernant la recette "REC". Contraintes : On ne peut utiliser simultanément plusieurs modes de fonctionnement.

La réponse "Y" à la question "SUPPRES MESS FIN DE CLE ?" permet de supprimer l'édition des messages "FIN OK" et "FIN NOK" des clés actions.

#### 4. 4. 5 STO - RST Store - Restore

Ces deux clés permettent de stocker (STO) et de restituer (RST) une phase de test. La clé STO doit être impérativement la première d'une phase de test. Toutes clés d'une même phase de test définies après la clé STO est enregistrée par le programme de test et peut être rappelée lors d'une nouvelle phase de test par la clé RST.

La clé RST doit être utilisée seule.

Exemple

```
DONNEZ VOS CLES
01 STO
02 101
03 102
04
DONNEZ VOS CLES
01
```



Les clés 101 - 102 sont enregistrées.

```
DONNEZ VOS CLÉS
01 RST
02
FIN 101 OK
FIN 102 OK
DONNEZ VOS CLÉS
01
```

Les clés exécutées sont RST - 101 - 102. Les clés 101 et 102 sont toujours enregistrées.

Une phase de test définie avec la clé ST0 est modifiée par une nouvelle définition de phase de test commençant par la clé ST0.

#### 4.5 L'APPEL OPERATEUR (BREAK)

Les programmes de test testent périodiquement si l'opérateur a fait un appel "BREAK" sur l'organe de dialogue. Le programme demande alors la définition d'une nouvelle phase de test. Le délai de réponse n'est pas nul et varie suivant les programmes.

#### 4.6 MESSAGES D'ERREUR

##### 4.6.1 Classes de Messages d'erreur

Le noyau de test SOLAR 16 offre 5 niveaux de messages d'erreur. Dans la conservation de test, il peut être demandé le mode d'édition des messages d'erreur. C'est un nombre pouvant aller de 1 à n ( $n \leq 5$ ) la valeur de n étant indiquée dans le libellé de la question.

Exemple : MODE ÉDIT MESS ERR (1 à 5) ?

Si l'on répond	
1 on obtient les messages de niveau	1
2	1 et 2
3	1 à 3
4	1 à 4
5	1 à 5

##### 4.6.2 Libellé des messages d'erreur

Le format des messages d'erreur issus des programmes de test est normalisé. Il est de la forme

```
ERR (xxx/yy) <----- Contenu du message----->
```

ou

xxx est le mnémonique de la clé où s'est produite l'erreur  
yy le numéro du texte d'explication dans la liste des messages d'erreur du manuel d'utilisation du programme de test.

Exemple : ERR (101/01) TIME-OUT

Cela veut dire qu'il y a erreur dans la clé 101, sur un problème de TIME-OUT et que pour comprendre ce qu'il s'est passé, il faut aller lire l'explication 01 de la liste des erreurs du manuel d'utilisation du programme de test.

Avertissement 2 messages de même libellé peuvent être produits dans des circonstances très différentes. Ils n'ont pas le même numéro d'explication.

#### 4.7 SURVEILLANCE PERMANENTE DES INTERRUPTIONS



Un programme de test n'utilise que un ou quelques niveaux parmi les 16 niveaux d'interruptions (0-15) de la série SPS 5. Ces niveaux du programme de test sont gérés par des tâches hardware dans le programme de test. Tous les autres niveaux non utilisés par le programme de test sont surveillés de façon permanente par le noyau de test. Si une interruption arrive sur ces niveaux, ce ne peut être qu'une interruption parasite; le noyau édite alors l'un des deux messages d'erreur suivants selon le niveau d'interruption. Puis le test continue de se dérouler de façon normale, sauf si l'interruption parasite est permanente.

- si l'interruption a eu lieu sur le niveau 0 (alarme) et si ce niveau est sans surveillance du noyau c'est-à-dire non utilisé par le programme de test.

ALARME 'X où X = numéro alarme

X =	0	mémoire inexistante
	1	protection DRPS ou CDA
	2	erreur de parité hors du mode debug
	3	instruction optionnelle inexistante
	4	instruction privilégiée
	5	RSQT, WAIT, QUIT sous-niveau hard
	6	alarme interprocesseur IPI
	7	STEP
	8	point d'arrêt (mode DEBUG)
	9	ACTD

- pour tous les autres niveaux surveillés.

IT PAR NIV 'X S/NIV 'YY

où

X = niveau d'IT parasite en hexadécimal  $1 \leq x \leq 15$

YY = sous-niveau d'IT parasite en hexadécimal  $0 \leq Y \leq 47$

sous-niveau normal 0-15, exceptions 16-47

Le programme de test surveille aussi les sous-niveaux inutilisés des niveaux qu'il utilise; il émet aussi des messages d'erreur quand arrive une interruption sur un sous-niveau qu'il n'utilise pas.

Quand une cause d'interruption parasite est permanente, le message est listé en permanence, le seul moyen pour arrêter cette cause d'interruption est de faire STOP - INITIALIZE - RUN.

RECIPROQUEMENT sur un module matériel fonctionnant correctement, mais dont on ne connaît plus le niveau et le ou les sous-niveaux d'interruption, en répondant de façon quelconque aux questions correspondantes du conversationnel de test, on peut, grâce à la surveillance par le noyau des interruptions, obtenir les renseignements cherchés.

## 5 ANNEXE - CHARGEMENT DES PROGRAMMES DEPUIS UNE BANDE MAGNETIQUE

### 5.1 BUT

Le système TESTEN permet la mise en oeuvre, des programmes de Test SPS 5 et éventuellement d'autres programmes depuis une Bande Magnétique 800 ou 1600 Bpi.

### 5.2 MOYENS NECESSAIRES

#### 5.2.1 Matériel

- Une configuration SPS 5 avec au minimum
- 32 K de mémoire vive, l'option scheduler
- Un périphérique de dialogue
- Un coupleur de Bande Magnétique
- Un dérouleur de Bande Magnétique 800 ou 1600 Bpi

#### 5.2.2 Logiciel

- Une Bande Magnétique TESTBM 800 ou 1600 Bpi suivant le dérouleur de Bande Magnétique utilisé.

#### 5.2.3 Documentation

- Le présent Manuel

### 5.3.1 Chargement et Lancement

- Monter la Bande Magnétique sur le Dérouleur et la mettre prête
- Bootstrapper la bande magnétique.

A ce niveau, le Bootstrap et le Programme TESTBM ont été chargés en mémoire

Le Programme TESTBM est lancé, il imprime une étoile sur l'organe de dialogue et passe en attente de clé

Pour TESTBM, une clé est un module activé par son mnémonique de 4 lettres.

A la fin d'exécution de ce module, le contrôle est rendu

- soit au Programme qui a été chargé,
- soit à TESTBM qui attend une nouvelle clé

EXEMPLE :

*CONF	Exécution de la clé CONF et retour à TESTBM
*REWI	Exécution de la clé REWI et retour à TESTBM
*TEST	Exécution de la clé TEST et lancement du Test

### 5.3.2 Clé CONF - Configuration

TESTBM est livré pour un Coupleur Bande Magnétique configuré de la manière suivante

ADR - Adresse coupleur	'18
HLW - Niveau HLW	14
CNL - Type de canal	HDC
ITN - Niveau IT canal	2
NPU - Numéro Processeur d'échange	1

Si le Coupleur utilisé correspond à cette configuration, il est inutile d'utiliser la clé CONF  
Dans le cas contraire :

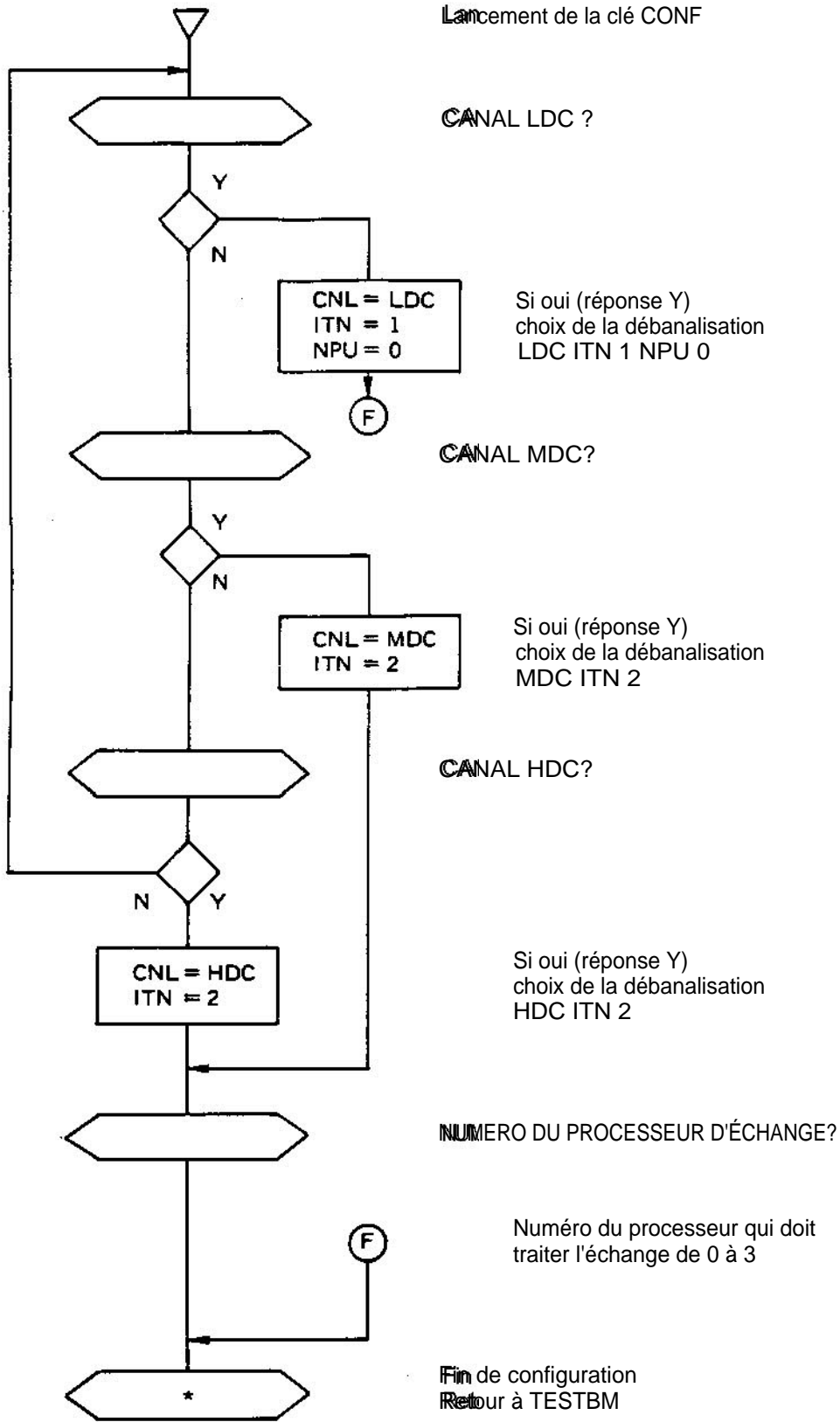
- Les paramètres ADR et HLW sont non modifiables
- Les paramètres CNL, ITN'et NPU peuvent être modifiés par la clé CONF

La configuration définie reste valable jusqu'à nouvelle utilisation de la clé CONF ou nouveau bootstrappage de la Bande.





Lancement de la clé CONF



### 5.3.3 Clé REWI - Rebobinage

Réseaux  
et systèmes  
d'information



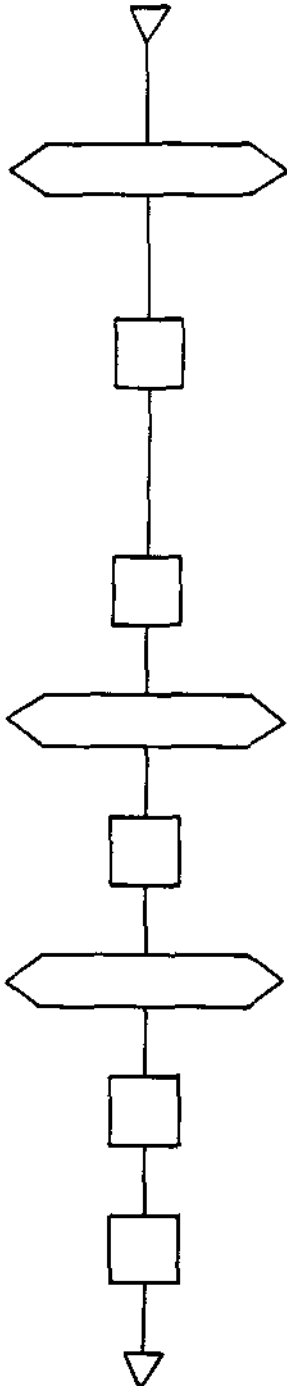
Cette clé lance un rebobinage sur la Bande Magnétique et redonne le contrôle à TESTBM.

On pourra ainsi dans certains cas bootstraper à nouveau la Bande sans intervention manuelle au niveau du Dérouleur.

5.2.1 CLÉ TEST - Lancement d'un Test



Cette clé permet de charger et de lancer un programme de Test disponible sur la Bande Magnétique.



Lancement de la clé TEST

TEST?

Donnez le nom mnémorique du test

TESTBM choisit un des trois noyaux en fonction du nom du test :

- TI ...    → Noyau TI
- ME ..    → Noyau ME
- CA    }    → Noyau Normal
- Autre }    → Noyau Normal

TESTBM recherche et charge le Noyau

EDITION DU LABEL DU NOYAU

TESTBM recherche et charge le test

EDITION DU LABEL DU TEST

TESTBM initialise la mémoire 4 et la mémoire 9, relais du Noyau et d'ASYNC

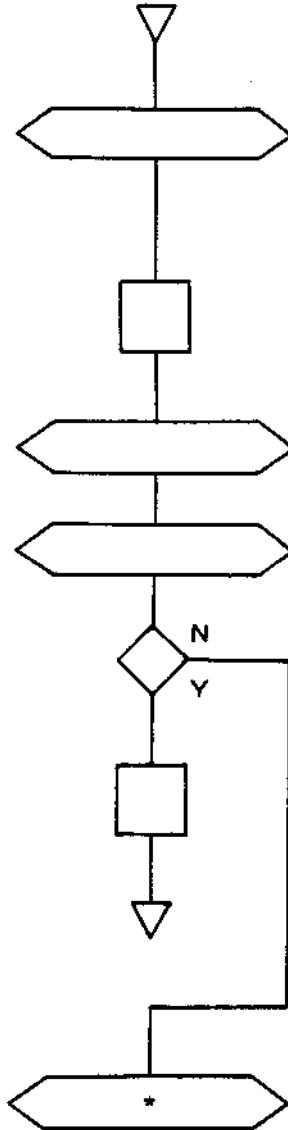
TESTBM lance le programme de test

Conversationnel du test

5.3.5 Clé PROG - Lancement d'un Programme



Cette clé permet de charger et de lancer un Programme disponible sur la Bande Magnétique.



Lancement de la clé PROG

PROG ?

Donnez le "Nom-Catalogue" du programme

TESTBM recherche et charge le programme demandé

EDITION DU LABEL DU PROGRAMME

RUN ?

Lancement du programme ? (Y ou N)

Si oui, TESTBM lance l'exécution du programme

Début d'exécution du programme

Si non  
Retour à TESTBM

Lancement d'un Test des instructions sur un SPS 5 avec Dérouleur en MDC 2 processeur 0.

- Bootstrapage de la bande

```
*CONF
LDC?N
MDC?Y
NUM. PROC. D'ECHANGE?0
*TEST
TEST?TI1

      NOYAU DE TEST DES INSTRUCTIONS
      1.158.000.02/...

      TEST DES INSTRUCTIONS
      1.158.200.02/...

HTR? Conversationnel du Test
```

### 5.3.7 Erreurs détectées

Le module TESTBM détecte un certain nombre d'erreurs soit au niveau Moniteur, soit au niveau des Clés

#### Erreurs au niveau moniteur (BOS-B + IOCS-M1)

ERB 00 X	Alarme X
ERB 04	Niveau/sous-niveau non géré :IT parasite
ERB 05	Time Out organe de dialogue : faire Break
ERB 06	Nom de clé incorrect
ERB 02	Défaut secteur : rebootstraper la bande
ERB 13	Défaut d'un périphérique DBM ou TTY
ERB 14	Appel opérateur : redonnez une clé

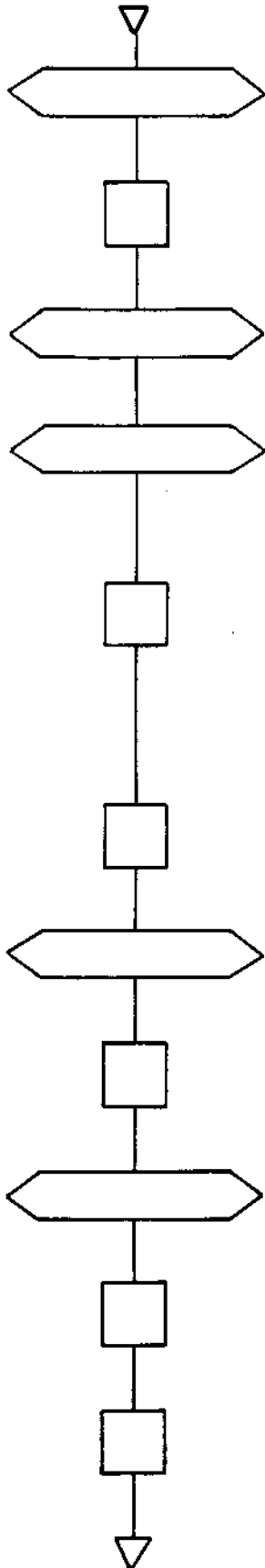
#### Erreurs au niveau des clés

ERREUR DIALOGUE!
Défaut périphérique ou erreur dans la syntaxe
DÉFAUT BANDE MAGNÉTIQUE!
Défaut périphérique résultant éventuellement de la non existence d'un test ou d'un programme demandé
DÉFAUT TABLE D'INDEX!
Informations incorrectes sur la bande magnétique

### 5.3.8 Changement de l'UC de traitement du SPS 5 bi-processeur



Le chargement des Programmes de test dans la mémoire de l'UC de traitement est fait par le frontal



Lancement de la clé PROG

PROG ?

BIBM-:S

Test BM cherche et charge le processeur BIBM-:S

RUN?

V (RC) lance l'exécution du processeur BIBM-:S

TEST ?

Donnez le nom mnémonique du test

TESTBM choisit un des trois noyaux en fonction du nom du test :

TI...	Noyau TI
ME ...	} Noyau ME
CA ...	
Autre ..	Noyau Normal

TESTBM recherche et charge le Noyau

EDITION DU LABEL DU NOYAU

TESTBM recherche et charge le test

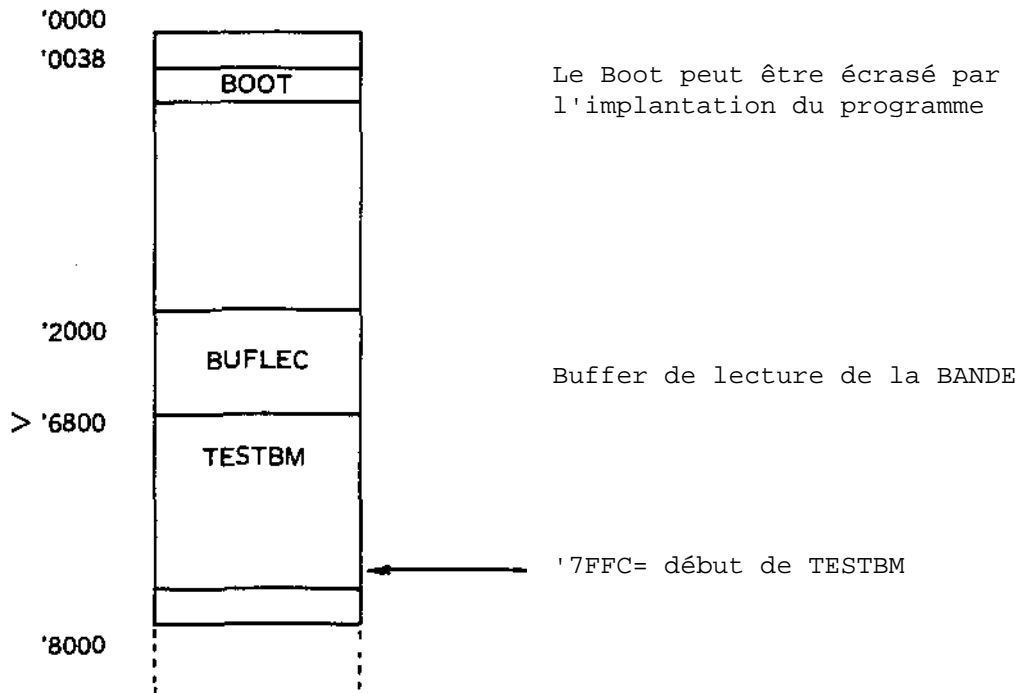
EDITION DU LABEL DU TEST

TESTBM initialise la mémoire 4 et la mémoire 9, relais du Noyau et d'ASYNC

TESTBM lance le programme de test

Conversationnel du test

### 5.3.9 Implantation de TESTBM



- TESTBM peut toujours être réactivé par l'adresse '7FFC tant qu'il n'a pas été écrasé
- Les programmes (Noyau. Test ou Autre..) lus sur bande sont alimentés dans BUFLEC, puis après analyse du descripteur sont transférés à leur adresse d'implantation.
- Contraintes :
  - Les Programmes doivent s'exécuter en Mode Maître Ils doivent avoir une Taille  $\leq 18$  K mots
  - Ils doivent s'implanter à une Adresse  $\leq '2000$ .
- Au lancement d'un Programme, la Mémoire d'adresse 'A contient la valeur '07FF. Fond de 32K.

### 5.3.10 Contraintes d'implantation des Tests



L'implantation en mémoire des Tests suit les règles énumérées ci-dessous.

- Le nom du Test permet de choisir un Noyau
  - Test TI... --> Noyau NOYATI
  - Test ME... --> Noyau NOYAME
  - Test CA... --> Noyau NOYAME
  - Autre Test --> Noyau NOYANO
- Les Noyaux et Tests sont des images mémoires chargées sur Bande Magnétique par FUP3 (IE≥40)
- Les Catalogues des fichiers sont
  - IF pour les Noyaux
  - IO pour les Tests
- Les Adresses d'implantations sont celles données par les descripteurs des fichiers
- Sauf modification indépendante de TESTBM, les Adresses d'implantation sont les suivantes

NOYATI	'170
NOYAME	'1000
NOYANO	'170
TI...	'8DO
ME..CA..	'1508
Autre.	'680
- Voir en 4-1- les autres contraintes liées à l'implantation de TESTBM.

### 5.3.11 Contraintes d'implantation des Programmes

L'implantation en mémoire des Programmes suit les règles énumérées ci-dessous

- Les Programmes sont des images mémoires chargées sur Bande Magnétique par FUP3 (IE ≥ 40)
- Ils sont identifiés par leur Nom et Catalogue qui peuvent être quelconques.
- Les Adresses d'implantation sont celles données par les descripteurs des fichiers.
- Voir en 4-1 les autres contraintes liées à l'implantation de TESTBM.



### 5.3.12 Liste de la bande magnétique



Il est possible de lister le contenu d'une Bande Magnétique TESTBM.

Pour cela, après avoir mis la Bande Magnétique en place on exécutera les commandes suivantes sous BOS-D.

```
/CALL FUP3
/U1 T1                Dérouleur utilisé
/LO LP
/TPIO REWI,U1
/TPIO FSFI,U1
/TPIO FSFI,U1
/TPIO FSFI,U1
/SCAN,U1, MTAP
/TPIO REWI, U1
/EOJ
```

### 5.3.13 Adjonction d'un test ou d'un programme sur la bande

Il est possible d'ajouter un Test ou un Programme à la fin d'une Bande Magnétique TESTBM.  
Pour cela, le Test ou le Programme doit être "buildé" pour constituer une Image Mémoire respectant les contraintes d'implantation précédemment définies.

Puis après avoir mis la Bande en place on exécutera les commandes suivantes sous BOS-D, avec :

(FIC-CT) = Nom-Catalogue de l'Image Mémoire.

```
/CALL FUP3
/U1 T1                Dérouleur utilisé
/U2 FU                FU support de l'Image Mémoire
/TPIO REWI, U1
/TPIO FSFI,U1
/TPIO FSFI,U1
/SKEOT,U1,MTAP
/FDUMP,FIC-CT,U2,DISK,U1,MTAP,2
/TPIO WTMA,U1
/TPIO BSFI,U1
/TPIO BSFI,U1
/TPIO BSFI,U1
/TPIO BSRE,U1
/FVERI,U1,MTAP,02,DISK
/TPIO REWI,U1
/EOJ
```

**VOS REMARQUES SUR CE DOCUMENT**

TITRE

NOYAU DE TEST

N DE REFERENCE

BULL MTS : 1 158 000 00 030 09

DATE

NOVEMBRE 1987

ERREURS DETECTEES

AMELIORATIONS SUGGEREES

\*→ Vos remarques et suggestions seront attentivement examinées.  
Si vous désirez une réponse écrite, veuillez indiquer ci-après  
Votre adresse postale complète.

NOM : ..... DATE.....

SOCIETE : .....

ADRESSE : .....

\*→ Remettez cet imprimé à un responsable Bull ou envoyez le  
directement à

**Bull S.A.**  
**CEDOC**  
20, rue Dieumegard  
93406 SAINT-OUEN CEDEX - FRANCE