

## I Mémoires débanalisées

0000	NS	N° de la tache soft en cours d'exécution.
0001	NL	N° de la tache soft qui nécessite une action du moniteur. (Complément de scheduling par tache soft 0)
0020	File ASTF	} 8 mots dont chaque bit mis à 1 indique que la tache de priorité $ASTF_i$ (0 à 127) a été demandé par l'instruction ARM. (le bit 0 du 1 <sup>er</sup> mot est le plus prioritaire)
0027		
0028		
002F	File ESTF	} 8 mots dont chaque bit mis à 0 indique que la tache soft de rang $i$ est masquée.
0030		
0037	File RSTF	} 8 mots dont chaque bit mis à 0 indique que la tache soft de rang $i$ nécessite un complément de scheduling. (toujours exécuté par la tache soft 0).

ASTF <sub>i</sub>	ESTF <sub>i</sub>	RSTF <sub>i</sub>	
1	1	1	Tache activable.
1	1	0	Tache éligible nécessitant un complément de scheduling.
1	0	0	Tache en attente non prête.
1	0	1	Tache en attente prête.

## II Tache soft 0 - Complément de scheduling

- La tache soft 0 est lancée par le scheduler lorsqu'une tache est éligible.  
(ASTF=1, ESTF=1, RSTF=0)
  - Les bits de la tache soft 0 (bit 0 du 1<sup>er</sup> mot de chaque file) doivent être:  
ASTF<sub>0</sub>=0, ESTF<sub>0</sub>=1, RSTF<sub>0</sub>=1
  - Le psoft va mettre: NS = '0000 (tache soft 0)  
NL = n n° de la tache soft ayant demandé le complément de scheduling.  
ASTF<sub>0</sub> = 1 tache soft 0 armée.
  - C'est le programmeur (dans la tache soft 0) qui mettra par programme le bit RSTF de la tache demandeuse à 1.
  - Le QUIT de la tache 0 remettra ASTF<sub>0</sub> à 0.
- \* Le complément de scheduling de la tache  $i$  est nécessaire lorsqu'au moment de son activation cette tache n'est pas présente en mémoire. Le rôle de la tache soft 0 sera d'appeler la tache soft  $i$  sur le disque par exemple.

Il doit toujours y avoir au moins une tache activable (3 bits à 1) dans les files du scheduler.



SPS 5

SCHEDULER

N° Document

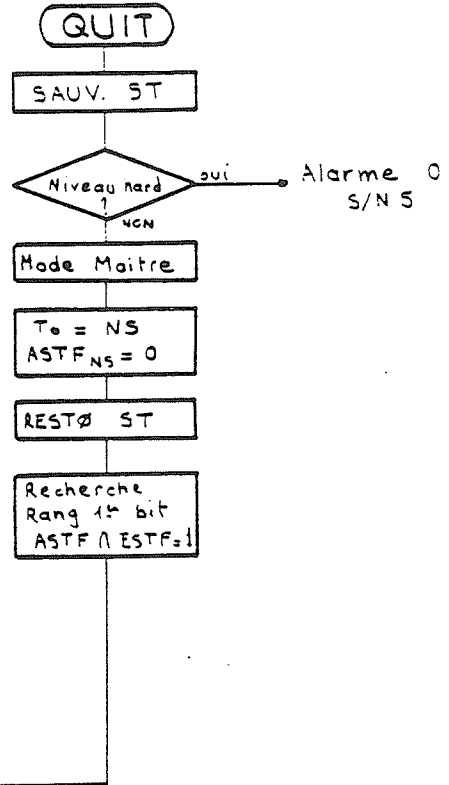
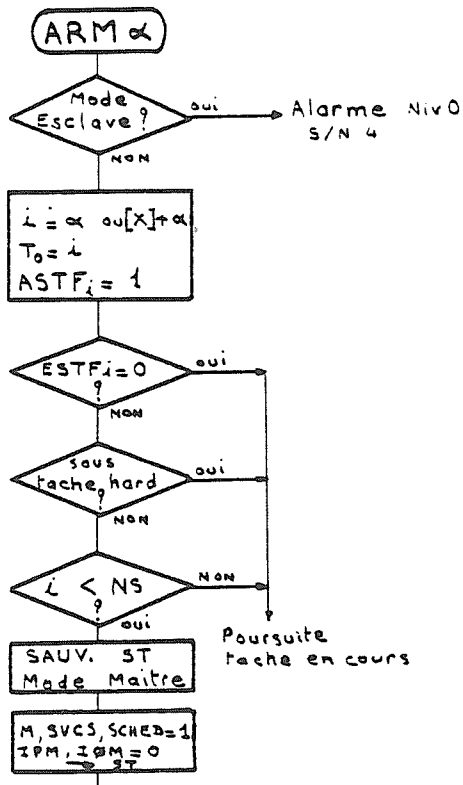
71 F7 31MS

Date

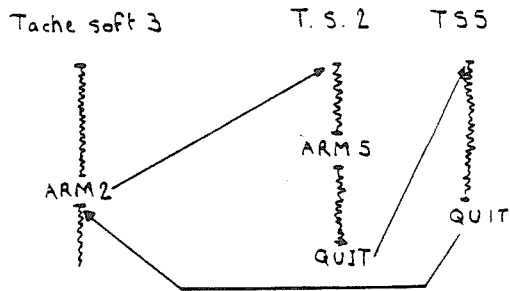
547

Page

P. 6.1



Exemple d'utilisation:

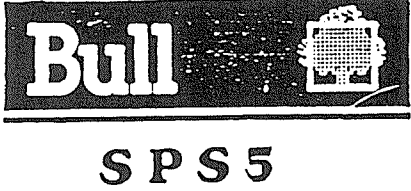


Pour compl. de scheduling.

Exécution tâche restaurée

- Une tâche soft est armée par une tâche hard ou soft (instruction ARM)
- Une tâche soft est toujours désarmée par elle-même (instruction QUIT)

SCHEDULER



N° Document

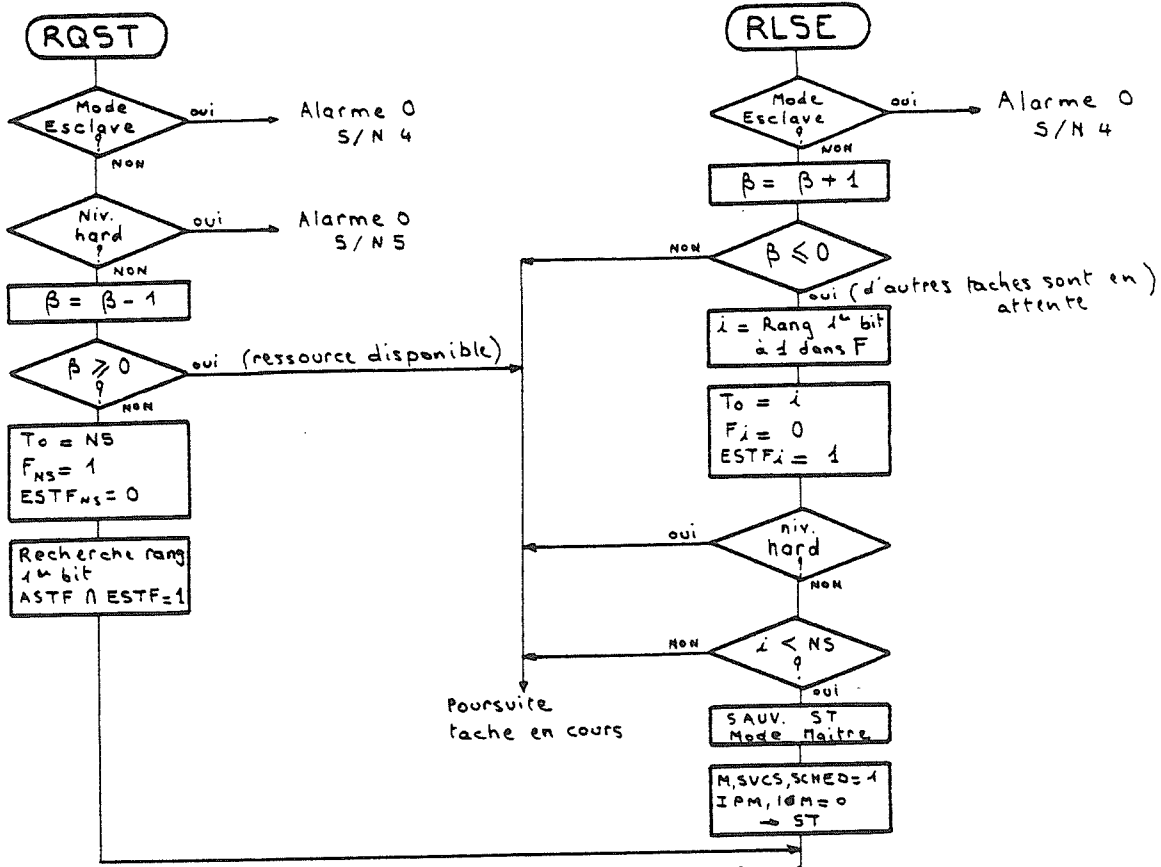
71 F7 31MS

Date

547

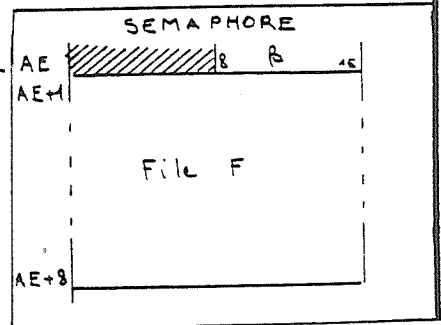
Page

P. 6.2

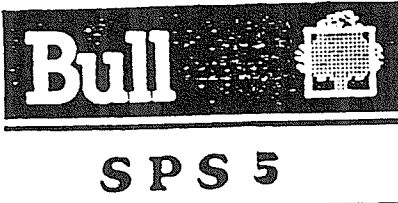


Sémaphore d'exclusion:

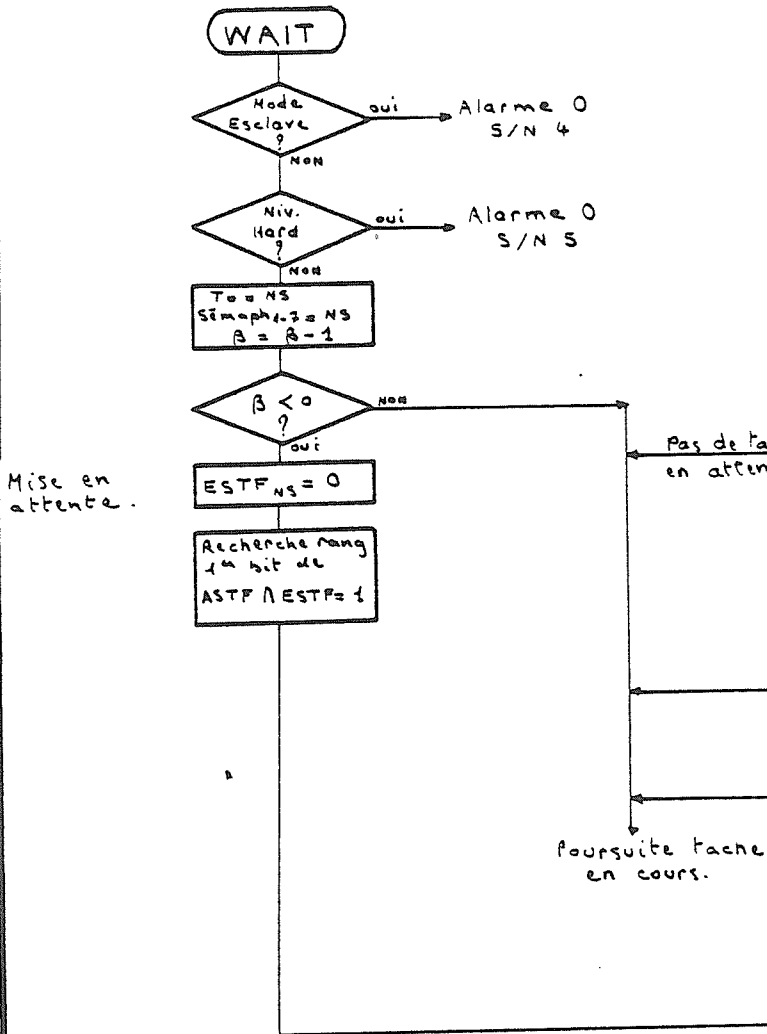
- Les sémaphores sont utilisés lorsque plusieurs taches peuvent être amenées à utiliser des moyens de travail en concurrence (périph, sous-programme ---). Ces moyens sont des ressources.
- Le nombre d'accès à la ressource correspond au nombre d'utilisateurs qu'elle peut satisfaire en même temps. Généralement ce nombre est égal à 1.
- Le compteur  $\beta$  du sémaphore doit être initialisé à une valeur qui détermine le nombre d'accès possibles (ex: 1)
- La file  $F$  mémorise les taches en attente voulant accéder à la ressource.
- La demande d'accès est faite par RQST.  $\Rightarrow \beta = \beta - 1$ 
  - Si  $\beta \geq 0 \Rightarrow$  Ressource disponible - Poursuite tache en cours.
  - Si  $\beta < 0 \Rightarrow$  le n° de priorité du demandeur est mémorisé dans la file  $F$  - Mise en attente - Lancement d'une autre tache soft.
- La libération est faite par RLSE  $\Rightarrow \beta = \beta + 1$ 
  - Si  $\beta > 0 \Rightarrow$  Pas de demandeur en attente. la tache se poursuit.
  - Si  $\beta \leq 0 \Rightarrow$  Lancement de la tache en attente la plus prioritaire.



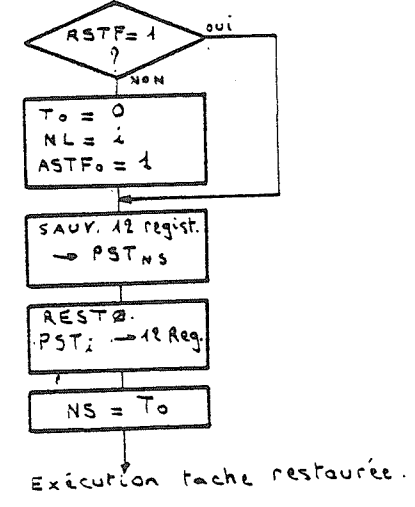
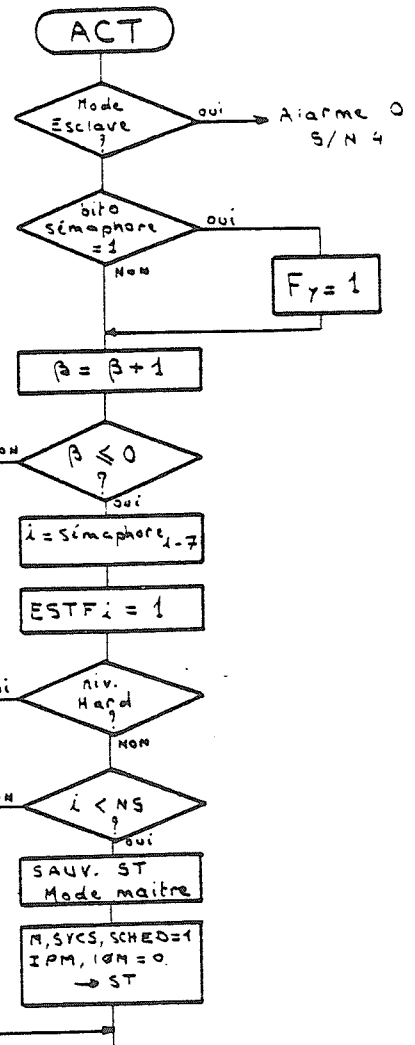
SCHEDULER



N° Document	Date	Page
71 F7 31MS	547	P. 6.3

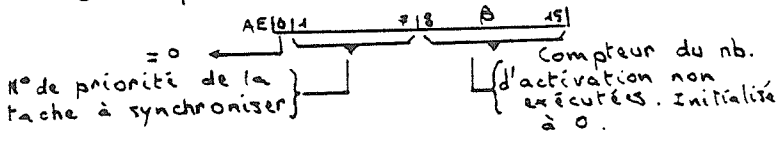


Mise en attente.



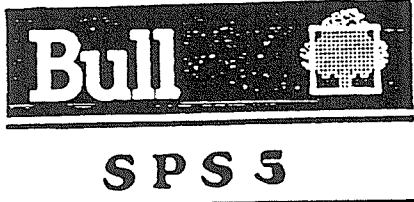
Sémaphore privé simple ou de synchronisation

- tâche se mettant en attente d'un événement positionné par une autre tâche.  
Le sémaphore est constitué d'un seul mot.



- Demande de synchro: WAIT  $\Rightarrow B = B - 1$
- si  $B \geq 0 \Rightarrow$  la tâche est poursuivie (l'évènement a été positionné.)
- si  $B < 0 \Rightarrow$  la tâche est mise en attente. Le scheduler est activé et lance la tâche soft activable la plus prioritaire.

SCHEDULER



N° Document

71 F7 31MS

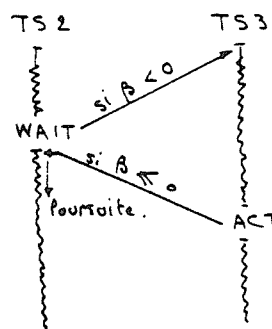
Date

547

Page

P. 6.4

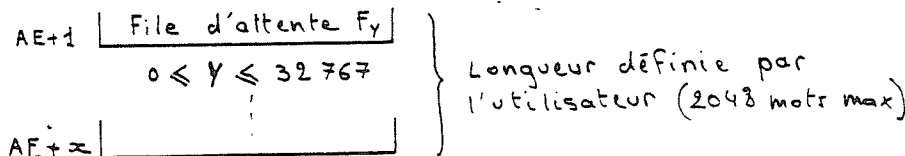
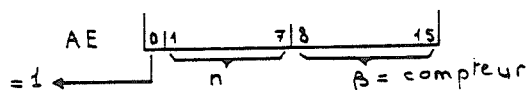
- Positionnement de l'évènement :  $ACT \Rightarrow \beta = \beta + 1$
- si  $\beta > 0 \Rightarrow$  il n'y a pas de tâche en attente et la tâche se poursuit.
- si  $\beta \leq 0 \Rightarrow$  La tâche de numéro  $n$  (dans sémaphore 1.7) est démasquée - Elle est activé si elle est plus prioritaire que la tâche en cours.



### Sémaphore privé avec File paramétrée

- Même fonctionnement que le sémaphore privé simple.
- Il permet en plus d'enregistrer lors du positionnement de l'évènement (instruction ACT) un numéro de paramètre donné dans le registre Y.

Sémaphore:



- Le sémaphore est associé à la tâche appelée.
- Lors du ACT le bit de rang donné par la valeur de Y est mis à 1 dans la File Fy.
- L'utilisateur doit effectuer la gestion de la File Fy, en particulier:
  - analyser les bits présents.
  - les remettre à 0 après utilisation.
- \* Le scheduler ne fait aucun contrôle sur la valeur du registre Y.