



**HAL**  
open science

# Conception d'un périphérique intelligent d'accès à un réseau télé-informatique : réalisation sur une architecture à microprocesseurs multiples

Roberto Saettone Summers

► **To cite this version:**

Roberto Saettone Summers. Conception d'un périphérique intelligent d'accès à un réseau télé-informatique : réalisation sur une architecture à microprocesseurs multiples. Modélisation et simulation. Institut National Polytechnique de Grenoble - INPG, 1979. Français. NNT : . tel-00289793

**HAL Id: tel-00289793**

**<https://theses.hal.science/tel-00289793>**

Submitted on 23 Jun 2008

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# THESE

*présentée à*

**Institut National Polytechnique de Grenoble**

*pour obtenir le grade de*

DOCTEUR INGENIEUR  
«Génie Informatique»

*par*

M. Roberto SAETTONE SUMMERS



CONCEPTION D'UN PERIPHERIQUE INTELLIGENT D'ACCES

A UN RESEAU TELE – INFORMATIQUE

REALISATION SUR UNE ARCHITECTURE A MICROPROCESSEURS MULTIPLES



Thèse soutenue le 11 Juillet 1979 devant la commission d'examen

L. BOLLIET                      Président

F. ANCEAU  
J.C. CHUPIN                      Examineurs

A. GUYOT  
H. ZIMMERMANN



# INSTITUT NATIONAL POLYTECHNIQUE DE GRENOBLE

Année universitaire 1977-1978

Président : M. Philippe TRAYNARD

Vice-présidents : M. René PAUTHENET

M. Georges LESPINARD

---

## PROFESSEURS TITULAIRES

MM. BENOIT Jean	Electronique - automatique
BESSON Jean	Chimie minérale
BLOCH Daniel	Physique du solide - cristallographie
BONNETAIN Lucien	Génie chimique
BONNIER Etienne	Métallurgie
* BOUDOURIS Georges	Electronique - automatique
BRISSONNEAU Pierre	Physique du solide - cristallographie
BUYLE-BODIN Maurice	Electronique - automatique
COUMES André	Electronique - automatique
DURAND Francis	Métallurgie
FELICI Noël	Electronique - automatique
FOULARD Claude	Electronique - automatique
LANCIA Roland	Electronique - automatique
LONGEQUEUE Jean-Pierre	Physique nucléaire corpusculaire
LESPINARD Georges	Mécanique
MOREAU René	Mécanique
PARIAUD Jean-Charles	Chimie - physique
PAUTHENET René	Electronique - automatique
PERRET René	Electronique - automatique
POLOUJADOFF Michel	Electronique - automatique
TRAYNARD Philippe	Chimie - physique
VEILLON Gérard	Informatique fondamentale et appliquée
* en congé pour études	

## PROFESSEURS SANS CHAIRE

MM. BLIMAN Samuël	Electronique - automatique
BOUVARD Maurice	Génie mécanique
COHEN Joseph	Electronique - automatique
GUYOT Pierre	Métallurgie physique
LACOUME Jean-Louis	Electronique - automatique
JOUBERT Jean-Claude	Physique du solide - cristallographie

.../...

MM.	ROBERT André	Chimie appliquée et des matériaux
	ROBERT François	Analyse numérique
	ZADWORNY François	Electronique - automatique

#### MAITRES DE CONFERENCES

MM.	ANCEAU François	Informatique fondamentale et appliquée
	CHARTIER Germain	Electronique - automatique
	CHIAVERINA Jean	Biologie, biochimie, agronomie
	IVANES Marcel	Electronique - automatique
	LESIEUR Marcel	Mécanique
	MORET Roger	Physique nucléaire - corpusculaire
	PIAU Jean-Michel	Mécanique
	PIERRARD Jean-Marie	Mécanique
	SABONNADIÈRE Jean-Claude	Informatique fondamentale et appliquée
Mme	SAUCIER Gabrielle	Informatique fondamentale et appliquée
M.	SOHM Jean-Claude	Chimie Physique

#### CHERCHEURS DU C.N.R.S. (Directeur et Maîtres de Recherche)

M.	FRUCHART Robert	Directeur de Recherche
MM.	ANSARA Ibrahim	Maître de Recherche
	BRONOEL Guy	Maître de Recherche
	CARRE René	Maître de Recherche
	DAVID René	Maître de Recherche
	DRIOLE Jean	Maître de Recherche
	KLEITZ Michel	Maître de Recherche
	LANDAU Ioan-Doré	Maître de Recherche
	MATHIEU Jean-Claude	Maître de Recherche
	MERMET Jean	Maître de Recherche
	MUNIER Jacques	Maître de Recherche

#### Personnalités habilitées à diriger des travaux de recherche (décision du Conseil Scientifique)

##### E.N.S.E.E.G.

MM.	BISCONDI Michel	Ecole des Mines St. Etienne (dépt. Métallurgie)
	BOOS Jean-Yves	Ecole des Mines St. Etienne (Métallurgie)
	DRIVER Julian	Ecole des Mines St. Etienne (Métallurgie)

.../...

MM. KOBYLANSKI André	Ecole des Mines St. Etienne (Métallurgie)
LE COZE Jean	Ecole des Mines St. Etienne (Métallurgie)
LESBATS Pierre	Ecole des Mines St. Etienne (Métallurgie)
LEVY Jacques	Ecole des Mines St. Etienne (Métallurgie)
RIEU Jean	Ecole des Mines St. Etienne (Métallurgie)
SAINFORT	C.E.N. Grenoble (Métallurgie)
SOUQUET	U.S.M.G.
CAILLET Marcel	Ecole des Mines St. Etienne (Chim. Min. Ph.)
COULON Michel	Ecole des Mines St. Etienne (Chim. Min. Ph.)
GUILHOT Bernard	Ecole des Mines St. Etienne (Chim. Min. Ph.)
LALAUZE René	Ecole des Mines St. Etienne (Chim. Min. Ph.)
LANCELOT Francis	Ecole des Mines St. Etienne (Chim. Min. Ph.)
SARRAZIN Pierre	Ecole des Mines St. Etienne (Chim. Min. Ph.)
SOUSTELLE Michel	Ecole des Mines St. Etienne (Chim. Min. Ph.)
THEVENOT François	Ecole des Mines St. Etienne (Chim. Min. Ph.)
THOMAS Gérard	Ecole des Mines St. Etienne (Chim. Min. Ph.)
TOUZAIN Philippe	Ecole des Mines St. Etienne (Chim. Min. Ph.)
TRAN MINH Canh	Ecole des Mines St. Etienne (Chim. Min. Ph.)

**E.N.S.E.R.G.**

MM. BOREL	Centre d'études nucléaires de Grenoble
KAMARINOS	Centre national recherche scientifique

**E.N.S.E.G.P.**

M. BORNARD	Centre national recherche scientifique
Mme CHERUY	Centre national recherche scientifique
MM. DAVID	Centre national recherche scientifique
DESCHIZEAUX	Centre national recherche scientifique



à mes parents,  
ma femme,  
mes enfants.





Je tiens à remercier,

Monsieur L. BOLLIET, Professeur à l'Université de Grenoble, qui m'a fait l'honneur de présider le Jury de cette thèse,

Monsieur F. ANCEAU, Maître de Conférences à l'Institut National Polytechnique de Grenoble, qui a bien voulu m'accepter au sein de l'Equipe de Recherche en Architecture d'Ordinateurs qu'il dirige, et qui m'a guidé dans la conception et réalisation de ce travail,

Monsieur J.C. CHUPIN, chercheur à CII, pour l'intérêt qui a montré dans le jugement de cet ouvrage,

Monsieur A. GUYOT, assistant à l'IMAG, pour les remarques et conseils qu'il m'a fait tout au long de ce travail,

Monsieur H. ZIMMERMANN, responsable de recherche en Informatique Répartie à l'IRIA, pour les observations et suggestions qu'il a apporté à cet ouvrage,

Messieurs Y. BEKKERS et P. FONTANILLE de l'atelier de Micro-informatique de l'IMAG pour leur aide et collaboration dans la génération de l'assembleur croisé pour le microprocesseur 8X300 en utilisant leur générateur d'assembleurs GAGE,

Monsieur A. BOUGHLAM, auteur du système de traitement de textes REDAK, à l'aide duquel je pu faire l'élaboration matérielle de cette thèse,

En général à tous les collègues de l'équipe avec qui j'ai eu des échanges fructueux.



Liste des Abreviations Utilisés

BD	-	Bloc de Demande
BR	-	Bloc de Réponse
BSC	-	Binary Synchronous Communication
MCC	-	Module Contrôleur de Communication
MIC	-	Module Interface Canal
MIMD	-	Multiple Instructions, Multiple Data
MST	-	Module Station de Transport
NCP	-	Network Control Program
PAV	-	Protocole d'Appareil Virtuel
PIAR	-	Périphérique Intelligent d'Accès au Réseau
SE	-	Station Externe
SR	-	Station Résiduelle
ST	-	Station de Transport
ST2	-	Station de Transport, version 2
UCB	-	Unité de Contrôle Banalisé



## TABLE DE MATIERES

INTRODUCTION

1 - Cadre du système PIAR . . . . .	7
2 - Présentation de l'ouvrage . . . . .	9

CHAPITRE I LES RESEAUX D'ORDINATEURS . . . . . 11

1 - Généralités . . . . .	13
.1 Liaison entre ordinateurs . . . . .	14
2 - Le Reseau CYCLADES . . . . .	16
.1 Le Réseau de communication CIGALE . . . . .	18
.2 Protocoles de communication . . . . .	22
.1 Protocole Noeud-Noeud . . . . .	25
.2 Protocole Hôte-Noeud . . . . .	27
.3 Protocole de Transport . . . . .	28
.4 Protocoles inter-applications . . . . .	29
3 - La Station de Transport . . . . .	30
.1 Adressage . . . . .	30
.2 Le service de Transport . . . . .	31
.3 Négotiation de session . . . . .	32
.4 Services de base . . . . .	32
.1 Multiplexage/démultiplexage . . . . .	33
.2 Fragmentation/ré-assemblage . . . . .	34
.5 Services additionels . . . . .	35
.1 Contrôle d'erreurs . . . . .	35
.2 Contrôle de flux . . . . .	36
.3 Adressage réduit . . . . .	37

CHAPITRE II IMPLEMENTATION DE LA LA STATION DE TRANSPORT . . 39

1 - Introduction . . . . .	41
2 - Implantation de la Station de Transport . . . . .	42
.1 Intégration au système d'exploitation . . . . .	42
.2 Implantation en mode usager . . . . .	43
.3 Implantation dans un ordinateur frontal . . . . .	45

3 -	Spécifications de réalisation de la ST . . . . .	47
.1	Les interfaces externes . . . . .	47
.2	Les Processus de la ST . . . . .	48
.1	Le processus Interface Abonné . . . . .	48
.2	Le processus d'Emission . . . . .	49
.3	Le processus de Réception . . . . .	49
.4	Le processus de Réveil . . . . .	50
.5	Le processus Réseau . . . . .	50
.6	Le Sélecteur . . . . .	50
4 -	Implémentation de la ST sur l'IBM 360/67 . . . . .	52
.1	Le système SYNCOP . . . . .	52
.2	La Station de Transport . . . . .	53
.1	Le processus LIGNE . . . . .	54
.2	Le processus RECEPTION . . . . .	54
.3	Le processus EMISSION . . . . .	55
.4	Le processus REVEIL . . . . .	56
.3	Interface avec les abonnés . . . . .	58
.1	Interface abonné interne . . . . .	58
.2	Interface abonné externe . . . . .	59
 <u>CHAPITRE III DECOMPOSITION DE LA STATION DE TRANSPORT</u> . . . . .		63
1 -	Introduction . . . . .	65
2 -	Décomposition de la ST . . . . .	68
.1	Interface avec les abonnés . . . . .	68
.2	Fragmentation/ré-assemblage . . . . .	69
.3	Division des fonctions . . . . .	70
3 -	La Station Résiduelle . . . . .	71
.1	Description de fonctions . . . . .	71
.2	Implémentation de la Station Résiduelle . . . . .	72
.1	Implémentation en mode usager . . . . .	72
.2	Implémentation intégrée au système . . . . .	74

4 - La Station Externe . . . . .	77
.1 Description fonctionnelle . . . . .	77
.1 Négociation de session . . . . .	77
.2 Fragmentation/ré-assemblage . . . . .	81
.3 Contrôle d'erreurs . . . . .	82
.4 Contrôle de flux . . . . .	82
.5 Temporisations . . . . .	83
.6 Télégrammes . . . . .	83
.2 Interface avec la Station Résiduelle . . . . .	83
.3 Implémentation de la Station Externe . . . . .	84
5 - Définition du Système PIAR . . . . .	86
.1 Description fonctionnelle . . . . .	86
.1 Station de Transport . . . . .	86
.2 Liaison avec le réseau . . . . .	87
.3 Liaison avec l'ordinateur hôte . . . . .	87
.2 Considérations de compatibilité . . . . .	89
.3 Considérations de débit . . . . .	89
<u>CHAPITRE IV LES ARCHITECTURES A PLUSIEURS MICROPROCESSEURS</u>	91
1 - Introduction . . . . .	93
2 - Types d'Architecture . . . . .	95
.1 Systèmes répartis . . . . .	95
.2 Systèmes multi-microprocesseurs . . . . .	97
.3 Couplage entre processeurs . . . . .	98
3 - Considérations de conception . . . . .	100
.1 Choix de l'architecture . . . . .	101
.2 La Structure d'interconnexion . . . . .	102
.3 Caractéristiques des processeurs . . . . .	103
4 - Le système PIAR . . . . .	105
.1 Segmentation du logiciel . . . . .	105
.2 Allocation des tâches . . . . .	106
.3 Modules du système . . . . .	106
.4 Communication entre modules . . . . .	107
.5 Structure d'interconnexion . . . . .	108
.6 Allocation des ressources . . . . .	109



<u>CHAPITRE V</u>	LE SYSTEME PIAR . . . . .	111
1 -	Hiérarchie du logiciel . . . . .	113
.1	Propagation de l'activité . . . . .	115
.2	Réalisation matérielle . . . . .	118
2 -	Ressources de communication . . . . .	120
.1	Le Bus parallèle . . . . .	120
.1	Allocateur de Bus . . . . .	121
.2	Algorithme d'allocation . . . . .	121
.3	Réalisation de l'allocateur de Bus . . . . .	126
.2	La mémoire commune . . . . .	131
.1	Allocateur de mémoire . . . . .	131
.2	Algorithme d'allocation . . . . .	132
.3	Implémentation de l'allocateur . . . . .	134
.4	Mécanisme d'allocation . . . . .	137
3 -	Synchronisation entre modules . . . . .	139
.1	Chemins de Synchronisation . . . . .	139
.2	Le lien MIC-MST . . . . .	141
.3	Le lien MST-MIC . . . . .	143
.4	Le lien MCC-MST . . . . .	145
.5	Le lien MST-MCC . . . . .	147
<u>CHAPITRE VI</u>	LES MODULES DU SYSTEME PIAR . . . . .	149
1 -	Le module Contrôleur de Communication . . . . .	151
.1	Description fonctionnelle . . . . .	151
.1	Gestion de la ligne physique . . . . .	151
.2	Gestion de la ligne logique . . . . .	152
.3	Gestion de l'interface avec les modules . . . . .	152
.2	Description du matériel . . . . .	153
.3	Description du logiciel . . . . .	155

2 - Le module Station de Transport . . . . .	158
.1 Description du matériel . . . . .	158
.2 Structure de la Station Externe . . . . .	160
.1 Le processus P-INTRES . . . . .	160
.2 Le processus P-EMIS . . . . .	161
.3 Le processus P-RECEP . . . . .	161
.4 Le processus P-REVEIL . . . . .	162
.5 Le processus MONITEUR . . . . .	162
3 - Le module Interface Canal . . . . .	164
.1 Introduction . . . . .	164
.2 Caractéristiques de l'UCB . . . . .	166
.3 Description fonctionnelle . . . . .	167
.1 Description du matériel . . . . .	167
.1 La partie contrôle . . . . .	169
.2 La partie interface canal . . . . .	170
.3 La partie interface dispositif . . . . .	170
.2 Description du logiciel . . . . .	171
CONCLUSION . . . . .	173
REFERENCES . . . . .	177
ANNEXE I Logiciel du module MCC . . . . .	185
ANNEXE II Logiciel du module MST . . . . .	193
ANNEXE III Logiciel du module MIC . . . . .	201
ANNEXE IV Rapport de réalisation du système PIAR . . . . .	221



## INTRODUCTION

### 1 - Cadre du système PIAR

Les options prises par les concepteurs du réseau CYCLADES obligent aux ordinateurs reliés de supporter un logiciel d'interface assez important. A notre avis, c'est d'ailleurs l'implémentation de ce logiciel qui constitue l'un des plus graves obstacles pour la connexion d'un ordinateur quelconque au réseau.

Dans cette optique, nous nous sommes penchés sur les problèmes de la connexion au réseau CYCLADES de l'ordinateur IBM 360/67 du Centre Inter-universitaire de Calcul de Grenoble. Après une première étude nous avons conclu qu'il serait très souhaitable d'implanter le logiciel d'interface au réseau sur un système frontal intelligent. C'est ainsi que nous avons conçu le système Périphérique Intelligent d'Accès au Reseau bâti autour d'une architecture à microprocesseurs multiples.

La conception du système PIAR est influencée par deux tendances modernes dans la conception des architectures des systèmes d'ordinateurs.

D'une part, la tendance à la décentralisation des fonctions mêmes de l'ordinateur central. Dans les nouvelles architectures, les fonctions principales sont soigneusement délimitées et ne communiquent qu'à travers des interfaces et protocoles bien définis.

Au même temps, on assiste à une migration de l'intelligence de traitement vers la périphérie. Les unités périphériques deviennent de véritables ordinateurs plus ou moins miniaturisés.

D'autre part, le développement explosif de la technologie des circuits intégrés à grande échelle (LSI) et à très grande échelle (VLSI) a permis la réalisation d'une unité centrale, ou même d'un ordinateur (unité centrale, mémoire de programme, mémoire de donnée, coupleur d'entrée/sortie), sur un seul circuit intégré, et ceci à des coûts extrêmement bas.

Les microprocesseurs monolithiques tendent à remplacer une grande partie des fonctions réalisées jusqu'à présent en logique cablée. Les automates sont généralement réalisés à l'aide de programmes temps réel inter-réagissant avec le milieu extérieur. Cette réalisation des algorithmes apporte des avantages supplémentaires, parmi lesquels :

- facilité de programmation, donc d'implémentation;
- souplesse pour introduire des modifications;
- facilité de la mise au point grâce à des outils perfectionnés;
- possibilité de réaliser des automates très complexes à l'aide d'un matériel relativement simple et standardisé.

Ces deux tendances se renforcent mutuellement et guident les tendances modernes de la conception architecturale des ordinateurs. Nous appelons ici ordinateur, non seulement l'ordinateur central dans sa conception classique, mais aussi le micro-ordinateur qui se trouve à l'intérieur de la plupart des périphériques modernes (télé-imprimeurs, écrans de visualisation, imprimantes, etc.) qu'ils soient "intelligents" ou non.

Le système PIAR est ainsi conçu autour d'une architecture à micro-ordinateurs multiples, chacun d'entre eux exécutant un sous-ensemble spécifique de la fonction à réaliser.

## 2 - Présentation de l'ouvrage

Nous présentons dans le chapitre I le réseau CYCLADES, les options prises par les concepteurs, leurs avantages et conséquences. Ensuite nous décrivons les différents niveaux de protocoles et en particulier la station de transport.

Le chapitre II décrit les différentes méthodes pour implanter la station de transport d'un ordinateur hôte. On discute leurs avantages et désavantages et en particulier on présente son implémentation sur l'IBM 360/67 du CICG.

Dans le chapitre III nous proposons une décomposition des fonctions de la SI en deux parties : une Externe, implanté sur un ordinateur frontal de communication; et une Résiduelle exécuté par l'ordinateur hôte. La politique suivie est de décharger le plus possible l'ordinateur hôte des fonctions d'interface au réseau.

Le chapitre IV fait une présentation sommaire des différentes types d'architecture à plusieurs microprocesseurs. On discute leurs avantages et inconvénients ainsi que des considérations de conception. Par la suite on décrit les choix et options prises dans la conception du système PIAR.

Le chapitre V décrit l'architecture logicielle et matérielle du système PIAR ainsi que les moyens de communication et synchronisation entre les modules.

Finalement, dans le chapitre VI nous présentons chacun des modules, leur architecture matérielle interne et la décomposition du logiciel en processus.



CHAPITRE ILES RESEAUX D'ORDINATEURS1 - Généralités

.1 Liaison entre ordinateurs

2 - Le Réseau CYCLADES

.1 Le Réseau de communication CIGALE

.2 Protocoles de communication

.1 Protocole Noeud-Noeud

.2 Protocole Hôte-Noeud

.3 Protocole de Transport

.4 Protocoles des applications

3 - La Station de Transport

.1 Adressage

.2 Le service de Transport

.3 Négociation de session

.4 Services de base

.1 Multiplexage/démultiplexage

.2 Fragmentation/ré-assemblage

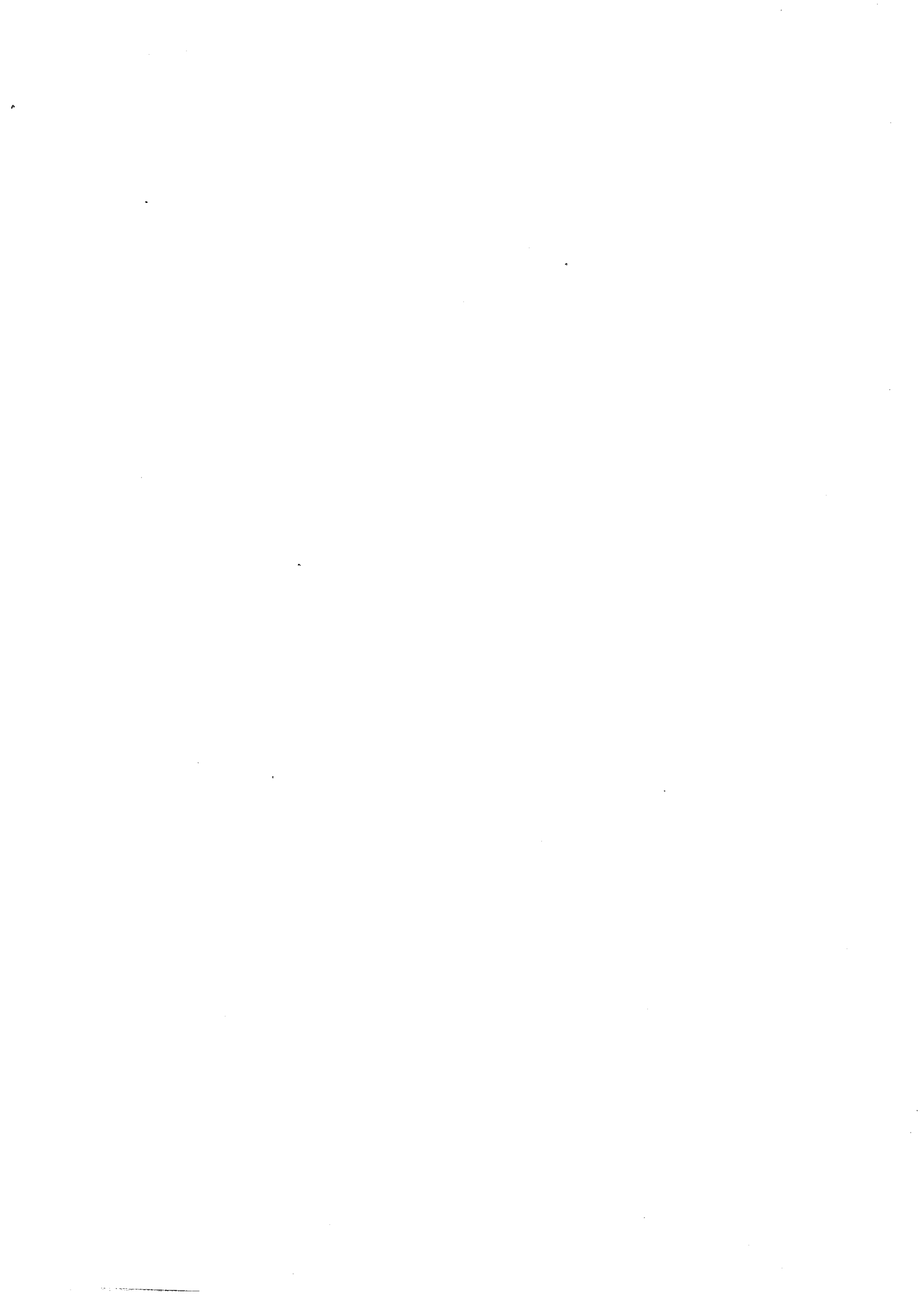
.5 Services additionels

.1 Contrôle d'erreur

.2 Contrôle de flux

.3 Adressage réduit





## I.1 - GENERALITES

Un réseau d'ordinateurs est un ensemble d'ordinateurs reliés entre eux par des voies de communication afin de partager leurs ressources matérielles et logicielles. Vis à vis d'un utilisateur, le réseau se présente comme une collection de ressources qu'il peut exploiter.

La puissance d'un réseau est directement proportionnelle au nombre d'ordinateurs connectés puisque le nombre de ressources disponibles pour l'ensemble des utilisateurs, augmente à chaque fois qu'un nouvel ordinateur est relié au réseau (mise en commun des ressources).

Un réseau d'ordinateurs peut être spécialisé, ou général, selon qu'il est orienté vers une application particulière ou non. Les buts n'étant pas les mêmes dans les deux cas.

Les réseaux spécialisés, tels que les réseaux bancaires ou ceux des compagnies aériennes, visent à une uniformisation des besoins et des systèmes d'exploitation afin de banaliser leurs ressources et de centraliser un certain nombre de services (comptabilité, statistiques, gestion de ressources, etc...).

Les réseaux d'ordinateurs généraux, à l'opposé des réseaux spécialisés, sont utilisés pour une grande variété d'applications. Ils regroupent généralement des ordinateurs et des ressources hétérogènes. En pratique, chaque utilisateur exploite un sous-ensemble de ressources particulières à son application, sans supposer qu'elles seront toujours les mêmes ni qu'elles auront toujours la même localisation géographique.

### I.1.1 Liaison entre Ordinateurs

Pour assurer la liaison entre les ordinateurs connectés au réseau, plusieurs solutions sont envisageables. Quelque soit le mode de couplage physique, on veut pouvoir établir entre eux des liaisons logiques leur permettant de communiquer deux à deux.

Si on établit une connexion physique pour chaque communication possible, on doit établir  $N*(N-1)/2$  lignes de connexion pour un réseau avec N ordinateurs (graphe complet). On observe que le nombre de liaisons augmente très rapidement et le réseau d'interconnexion devient très complexe. De plus, le taux d'utilisation de chaque liaison est très faible et le coût total d'exploitation très élevé. On note aussi que chaque ordinateur doit gérer (N-1) liaisons et que pour chaque ordinateur ajouté au réseau on doit établir N nouvelles liaisons.

Une solution, qui à première vue semble idéale, est d'utiliser le réseau téléphonique commuté. En effet, celui-ci permet d'établir une connexion entre une paire d'ordinateurs sur demande et seulement pendant le temps nécessaire. Les lignes du réseau commuté sont malheureusement, très mal adaptées à la transmission de données et présentent plusieurs inconvénients. Entre autres, les délais introduits par le temps d'établissement de la connexion, une largeur de bande insuffisante et une faible fiabilité due aux bruits électriques introduits par les centres de commutation (souvent électro-mécaniques), sans parler du risque de déconnexion pendant la transmission des données.

Il est donc évident qu'il est préférable d'utiliser des lignes dédiées en permanence au réseau pour la communication entre les ordinateurs reliés. Le problème est le coût élevé des lignes louées. Une solution consiste à ne pas utiliser des liaisons point-à-point entre chaque paire d'ordinateurs, mais de créer un outil de transport commun à tous les ordinateurs participants. Cet outil de transport doit pouvoir acheminer

simultanément tout les échanges possibles entre toutes les paires d'ordinateurs. L'outil de transport se présente aux ordinateurs comme une boîte noire. C'est à dire que les informations qui lui sont transmises sont livrées à destination sans altération (d'une façon transparente).

Dans le cas général, l'outil de transport en commun est un réseau de communication de forme maillé composé de noeuds reliés entre eux par des lignes téléphoniques louées. Ceci ne veut pas dire que chaque noeud est relié directement à tous les autres. En général, on détermine la topologie du réseau de communication en tenant compte des ressources de communication disponibles, de la capacité de transmission à assurer, de la fiabilité à obtenir et, bien sûr, du coût. Ceci peut être optimisé en utilisant plusieurs méthodes (1), en particulier en se basant sur des concepts de la théorie des graphes (2).

Chaque ordinateur participant est relié à au moins un noeud du réseau de communication. Les noeuds sont eux-mêmes des ordinateurs spécialisés, en général des mini-ordinateurs, orientés vers la transmission de données.

## I.2 - LE RESEAU CYCLADES

Le réseau d'ordinateurs CYCLADES est un réseau né en 1972 (3) qui couvre une grande partie du territoire Français. Il est connecté à d'autres réseaux américain et européens (4). CYCLADES est un réseau hétérogène puisqu'il relie des calculateurs et des terminaux de marques et des modèles différents. Il est aussi un réseau général parce qu'il n'est pas dédié à une application particulière mais relie entre eux des centres de calcul d'université et des centres de recherche qui l'utilisent pour des applications tout à fait différentes.

Le réseau CYCLADES supporte plusieurs applications qui ont déjà dépassé le stade expérimental et qui sont actuellement en exploitation (5). Les principaux services offerts mettent à la disposition des utilisateurs les systèmes de temps partagé et traitement par lots des gros ordinateurs connectés. D'autres applications sont en cours de développement tel que le projet de base de données réparties (6), le développement d'un langage de commande pour le réseau (7), et des concentrateurs microprogrammés permettant de connecter des terminaux directement au réseau (8).

La structure du réseau CYCLADES est caractérisée par :

- l'indépendance entre les ordinateurs connectés et l'outil de transport qui sert à les relier,
- la hiérarchisation des différents niveaux de logiciel et de matériel,
- la simplicité des traitements dans les cas simples.

De ce fait, CYCLADES peut être décrit comme étant constitué de deux niveaux bien définis :

- Un réseau de traitement, formé par tous les calculateurs participant au réseau, qui mettent à la disposition des utilisateurs leurs ressources matérielles et logicielles.
- Un réseau de communication qui s'occupe de transmettre d'un point à un autre des paquets d'information avec un taux d'erreurs très réduit.

Le premier niveau s'adresse au deuxième comme à une boîte noire. Chaque ordinateur y dépose et retire des paquets d'information sans se soucier ni du routage ni de la commutation des paquets dans le réseau de communication. Le réseau de communication accepte et délivre des paquets d'information sous forme transparente; c'est-à-dire sans tenir compte de la représentation, de la structure, ou du format des données transmises. Un paquet est livré à destination sans aucune altération ou modification.

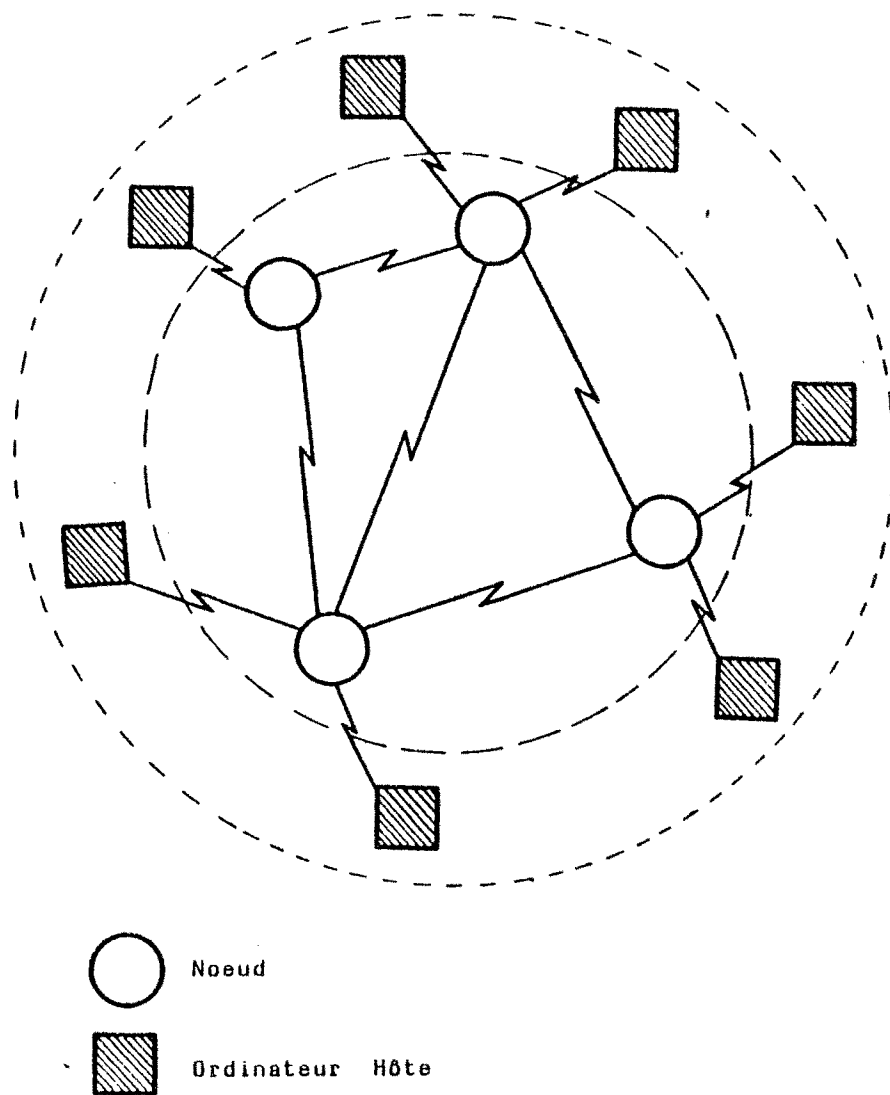


Fig. I-1 Les deux niveaux du réseau CYCLADES

### I.2.1 Le Réseau de Communication CIGALE

Le réseau de communication de CYCLADES, appelé CIGALE, est composé de mini-ordinateurs CII Mitra 15, reliés entre eux par des lignes téléphoniques louées formant un réseau maillé. Ces mini-ordinateurs constituent les noeuds auxquels sont reliés les ordinateurs participants (16).

Un noeud exécute les fonctions de commutation. Il délivre aux ordinateurs qui lui sont attachés, les messages qui leur sont destinés et retransmet à ses noeud adjacents les messages destinés aux autres.

CIGALE est un réseau de communication du type stockage et retransmission (store & forward). Dans chaque noeud, chaque message reçu est rangé dans un tampon de mémoire locale et est ensuite retransmis vers un autre noeud ou vers l'ordinateur destinataire si celui-ci lui est attaché. Dans les deux cas, le noeud attend un accusé de réception et ne libère pas le tampon tant qu'il ne l'a pas reçu.

CIGALE utilise la technique de commutation de paquets. Un paquet est un message d'information de longueur réduite (au maximum 255 octets) transporté par le réseau de communication. Un paquet remis à un point d'entrée du réseau de communication est transmis de noeud en noeud jusqu'à son point de destination.

La commutation de paquets établit une liaison virtuelle entre deux correspondants de façon telle que les circuits et les ressources nécessaires à la transmission peuvent être utilisées simultanément par d'autres liaisons.

La limitation de la longueur des paquets, diminue le délai de transit du paquet pour traverser le réseau de communication et permet d'avoir des mémoires de stockage moins importantes dans les noeuds. De plus, en cas d'erreur de transmission entre les noeuds, le volume d'information à retransmettre est faible.



L'itinéraire que doit suivre par un paquet dans le réseau CIGALE n'est pas pré-établi ni figé. Un mécanisme de routage adaptatif est appliqué de façon à tenir compte de la disponibilité des ressources de transmission des différentes voies du réseau. Ceci a pour conséquence que des paquets expédiés dans un certain ordre par un ordinateur participant, peuvent, éventuellement, être reçus par le destinataire dans un ordre différent.

A l'heure actuelle, le réseau de communication CIGALE est composé par huit noeuds de mini-ordinateurs CII Mitra 15. Un centre de contrôle surveille en permanence l'état du réseau. Chaque noeud peut être observé et téléchargé à partir d'un autre noeud autorisé, généralement le centre de contrôle.

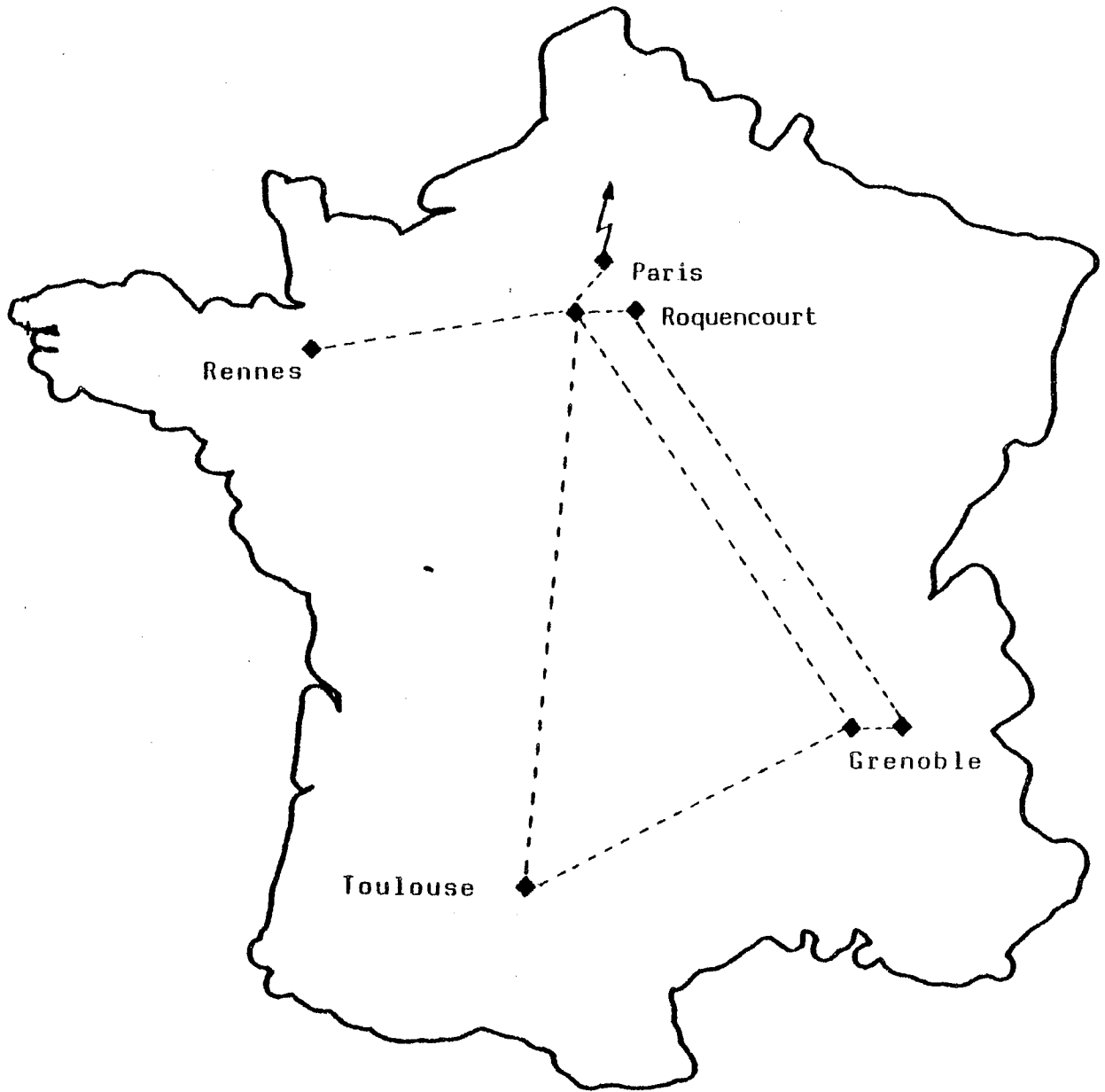


Fig. I-2 Topologie du Réseau CYCLADES

Contrairement au réseau de communication du réseau ARPA (9), les fonctions assurées par CIGALE se limitent au transfert d'un paquet depuis l'origine jusqu'à sa destination. Ceci simplifie la structure interne et les fonctions de chaque noeud en laissant les traitements plus complexes aux protocoles de plus haut niveau implementés dans les ordinateurs connectés.

Ces traitements sont :

- la fragmentation à l'origine des messages longs en paquets de taille réduite,
- le ré-assemblage des paquets reçus pour reconstituer le message original,
- la reprise sur perte des paquets ou des messages,
- l'élimination des paquets ou des messages doubles,
- le séquençement des paquets et des messages.

### 1.2.2 Protocoles de Communication

Lorsque deux partenaires veulent dialoguer entre eux, il faut que ce dialogue soit régi par un ensemble de règles de façon que la conversation soit cohérente.

On distingue deux niveaux de définition de ces règles:

- la procédure de communication qui décrit le format des messages,
- le protocole qui décrit la structure de son contenu.

Une procédure de communication est un ensemble de conventions ou de règles établies entre deux (ou plusieurs) correspondants leur permettant d'établir un dialogue cohérent. Lorsque la conversation se déroule sur une ligne téléphonique, ces conventions sont qualifiées de procédure de transmission.

Une procédure de communication est composé par :

- un alphabet qui décrit les caractères connus par les partenaires et, en particulier, les caractères de contrôle utilisés,
- une grammaire qui décrit les règles de composition des messages de données et de contrôle,
- un ensemble de règles qui déterminent l'enchaînement des messages échangés et permettent de détecter et de reprendre les erreurs dues à la perte ou à la duplication des messages.

Ces dernières peuvent être représentées à l'aide d'un automate d'états finis (11), ce qui facilite la représentation, la vérification et la compréhension du protocole.

Nous avons mentionné précédemment que le réseau CYCLADES est caractérisé par la hiérarchisation des protocoles et l'indépendance des différents niveaux. Ainsi nous trouvons dans le réseau CYCLADES 4 niveaux de protocoles :

- protocole de communication Noeud-Noeud,
- protocole de communication Hôte-Noeud,
- protocole de transport,
- protocole d'application.

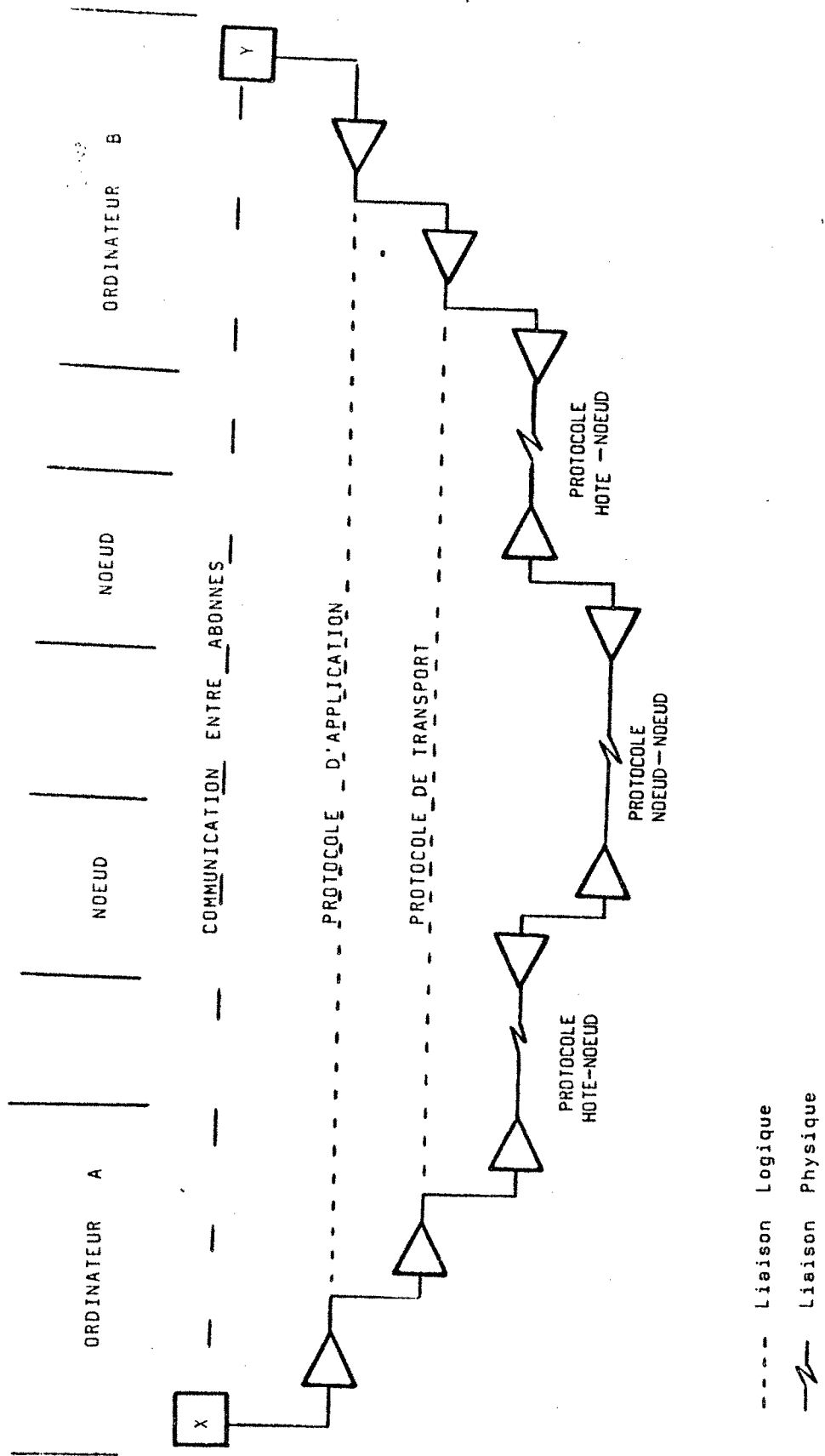


Fig. I-3 Les niveaux de protocoles de CYCLADES

### 1.2.2.1 Protocole Noeud-Noeud

Les noeuds de CIGALE communiquent entre eux suivant une procédure full-duplex, bi-point, particulière au réseau, appelé MV8 (10).

A la différence de la procédure TMM des ordinateurs CII-HB, MV8 permet une anticipation de 8 paquets à l'émission. Pour cela, elle met en jeu le principe de 8 canaux virtuels sur une même ligne physique.

Dans la procédure MV8, les paquets ne sont pas traités indistinctement, mais ils sont classés suivant leur longueur :

- Classe A, plus de 127 caractères,
- Classe B, de 5 à 127 caractères,
- Classe C, moins de 5 caractères.

De plus, les paquets correspondant à différentes classes ne sont pas envoyés indistinctement sur les 8 canaux logiques. Chaque classe de paquets est envoyée sur un sous-ensemble fixe de canaux. Ainsi la classe A est réservée aux canaux 2, 4 et 6; la classe B aux canaux 1, 3, 5 et 7; et la classe C à tous les canaux.

Les paquets en attente d'émission sont organisés en files. Il y a une file pour chacune des 3 classes de paquets. Les paquets dans une file sont gérés dans l'ordre d'arrivée: premier arrivé-premier servi. Par contre la file de la classe C est prioritaire par rapport aux files A et B. Ceci parce que, en général, les paquets de petite longueur correspondent à des messages de contrôle.

Chaque canal logique permet l'expédition d'un paquet à la fois, numéroté cycliquement modulo 2. Une fois le paquet émis, le canal est considéré occupé jusqu'à la réception de l'acquittement de ce paquet. Ceci est valable pour les 8 canaux, ce qui permet une anticipation générale de 8 paquets.

En raison de la distribution des canaux par classes, le taux d'anticipation est au maximum de 3 pour la classe A, de 4 pour la classe B et de 8 pour la classe C.

A chaque canal est associé un bit d'état indiquant s'il est occupé ou non, et un bit indiquant le numéro du dernier paquet émis. De plus, chaque canal possède une temporisation armée à l'émission d'un paquet. Si la temporisation échoue, le paquet est retransmis par le même canal et avec le même numéro.

En émission, un octet d'acquittement est ajouté à chaque paquet. Dans cet octet, le bit "i" correspond au numéro du dernier paquet bien reçu sur le canal "i". Cette méthode d'acquittement, suppose qu'il existe un trafic de paquets dans les deux sens. En absence de trafic dans un sens, un paquet vide est envoyé, comportant seulement l'octet d'acquittement.

Une deuxième version de MV8 a été écrite dans un but d'améliorer ses performances en éliminant la contrainte de l'allocation statique des canaux aux classes de paquets. Dans cette deuxième version, appelé MV8-CS, on autorise l'envoi de paquets d'une classe (A ou B) sur un canal destiné à l'autre classe si ce canal est libre et si la file de la classe propriétaire du canal est vide.

MV8-CS apporte une amélioration des performances par un facteur de 3 par rapport à MV8 dans le cas de trafic non équilibré (14).

#### 1.2.2.2 Protocole Hôte-Noeud

La communication entre les noeuds de CIGALE et les ordinateurs participants, est faite par des liaisons série en mode synchrone, "full-duplex". Ces liaisons sont gérées par la procédure de transmission TMM-UCIG. La seule exception est la liaison avec l'ordinateur IBM 360/67 du Centre Inter-Universitaire de Calcul de Grenoble, qui est géré par la procédure BSC d'IBM.

La procédure de communication fournit un ensemble de règles élémentaires pour le contrôle d'une ligne de transmission de données. Elle définit un format commun et une syntaxe pour tous les messages, ainsi que le séquençement des messages de contrôle, afin de détecter et récupérer les erreurs possibles de transmission. La validité des données est contrôlée à l'aide d'un code cyclique redondant (CRC) généré et concaténé à la fin de chaque message.

La procédure TMM-UCIG, est un sous-ensemble compatible de la procédure orientée octets, TMM, employé par les machines CII-HB. Celle-ci a été définie dans le but de simplifier au maximum les échanges d'information nécessaires au dialogue (12), tout en conservant la compatibilité avec la procédure TMM. En particulier tout ce qui est lié au problème de contention, a été éliminé. Ce problème étant inexistant dans CIGALE puisque les communications se font en mode "full-duplex".

La procédure TMM-UCIG est du type "envoi et attente", (send & wait). C'est à dire que l'émetteur envoie un paquet et se place en attente du message d'acquiescement de la part du récepteur. Tant qu'il ne l'a pas reçu, il n'envoie pas un autre paquet. Du fait qu'aucune anticipation n'est faite on peut considérer que la communication est faite en mode pseudo "half-duplex".



Pour éviter un blocage possible dû à une erreur de transmission sur la ligne, l'attente d'acquiescement est protégée par une temporisation.

Ce manque d'anticipation est très restrictif et des études par simulation (13) ont montré que dans le meilleur des cas, le protocole ne peut pas dépasser les 2/3 de la capacité de transmission de la ligne.

### I.2.2.3 Le Protocole de Transport

Le protocole de transport est un protocole dit de "bout-en-bout" qui gère les échanges entre l'origine et la destination des messages sans tenir compte des protocoles de communication de CIGALE. Il est implanté sur les ordinateurs hôtes reliés au réseau CYCLADES, et constitue une couche de logiciel appelée Station de Transport. Tout ordinateur participant possède une Station de Transport qui réalise l'interface entre les utilisateurs locaux et le réseau.

Le protocole de transport est donc un protocole de haut niveau, qui accomplit un ensemble de fonctions :

- détection et récupération des erreurs dues au réseau de communication CIGALE (séquencement, perte, duplication des messages);
- multiplexage de plusieurs dialogues entre les utilisateurs locaux et distants par la liaison Hôte-Noeud;
- simplification de l'accès au réseau par les programmes utilisateurs.

Le protocole de transport du réseau CYCLADES a comme équivalent le "Network Control Program", NCP du réseau ARPA. Toutefois, ce dernier ne comporte pas les fonctions d'erreurs sur les messages et délègue un certain nombre de tâches aux noeuds du réseau de communication. Ceci simplifie l'interface au réseau sur les calculateurs participants au prix de compliquer les tâches et l'architecture des noeuds. Le choix dans le réseau CYCLADES a été différent en profitant des expériences du réseau ARPA. On a simplifié les tâches et l'architecture du réseau de communication en reléguant les fonctions de haut niveau aux ordinateurs participants.

#### 1.2.2.4 Les Protocoles inter-Applications

Un protocole d'application ou protocole utilisateur est un ensemble de règles de dialogue régissant les échanges des messages entre les utilisateurs d'une application particulière. On appelle application, l'exploitation d'un ou plusieurs des services du réseau par un ou plusieurs utilisateurs ou correspondants géographiquement répartis.

Certains protocoles d'application ont été définis d'une façon standard pour l'ensemble du réseau, tels que le protocole d'appareil virtuel PAV, le protocole de transfert de fichiers et le protocole de soumission de travaux à distance (Remote Job Entry).

En particulier, le protocole d'appareil virtuel est très intéressant car il présente les différents types, marques et modèles de terminaux connectés au réseau d'une façon standard. Une application ou un utilisateur s'adresse à un terminal standard dit "appareil virtuel" sans se soucier de ses caractéristiques réelles. L'interface entre le protocole d'appareil virtuel et le terminal réel est fait localement.

### I.3 - LA STATION DE TRANSPORT

La station de transport (ST), est une couche de logiciel entre les utilisateurs locaux et le système de communication du réseau. Elle peut être décrite comme un ensemble de fonctions de haut niveau qui permettent aux utilisateurs locaux de dialoguer avec des partenaires distants sans être concernés par les problèmes normalement associés à un flux de données. La station de transport locale dialogue avec les stations distantes en assurant une liaison de communication fiable. La station de transport est implantée sur chaque ordinateur participant au réseau.

Nous décrivons par la suite, très brièvement, les principaux services offerts par la deuxième version de la Station de Transport du réseau CYCLADES, appelée ST2 (15).

#### I.3.1 Adressage

A l'intérieur du réseau, chaque station de transport possède une adresse unique (ST-AD) qui l'identifie pleinement. Cette adresse est utilisée par le réseau de communication CIGALE pour le routage des paquets destinés à une station de transport. D'autre part, tout utilisateur doit être abonné au réseau pour avoir accès à ses services.

Un abonné (AB) a accès au réseau au moyen d'une porte (PT). Chaque abonné peut avoir une ou plusieurs portes par lesquelles il peut dialoguer avec ses partenaires distants. Dans une station de transport, chaque porte a une identification unique (PT-ID) qui l'identifie pleinement. De ce fait, une porte a une identification unique dans le réseau par le couple : adresse de station de transport, identification de porte (ST-AD,PT-ID).

Un flot est le dialogue entre deux abonnés qui communiquent entre eux, chacun à travers d'une porte. Puisque chaque porte est identifiée de façon unique dans le réseau, l'identification du flot correspondant (FL-ID) est faite d'une manière unique par la combinaison des adresses des deux portes.

Chaque message échangé par les deux partenaires sur un flot comporte l'identification de porte origine et l'identification de porte destination. Un flot peut être vu comme la liaison entre deux correspondants par une voie virtuelle.

### I.3.2 Le Service de Transport

L'ensemble des stations de transport implantées dans les différents ordinateurs participants, constitue la machine de transport du réseau CYCLADES. Cette machine fournit une interface simple entre les utilisateurs, leur permettant d'échanger des messages.

Un utilisateur est une entité susceptible de recevoir ou d'émettre des informations du, ou vers, le réseau. Ainsi on considère comme utilisateur non seulement la personne qui accède au réseau à travers un terminal, mais aussi le terminal lui-même, les programmes d'application et les différentes ressources du réseau.

La machine de transport transmet l'information d'une façon transparente entre l'origine et la destination avec un taux d'erreurs nettement plus faible que celui du réseau de communication ( $\ll 10 E-10$ ).

### I.3.3 Négotiation de Session

Un flot peut être ouvert à la demande d'un abonné local ou d'un partenaire distant. A l'ouverture du flot, le demandeur spécifie les services qu'il veut voir exécuter sur le flot. Le flot n'est ouvert que si les deux abonnés sont d'accord sur les services à exécuter suivant un protocole de négociation de session. De même, la session est fermée à la demande d'un des deux partenaires ou à la détection d'une erreur irrécupérable par l'un d'entre eux.

Les sequences d'ouverture et de fermeture de flot sont faites à l'aide de deux commandes : FL-INIT et FL-TERM.

### I.3.4 Services de Base

En service de base, les informations émises par un abonné sur un flot sont traitées par les stations de transport de façon à les mettre dans un format compatible avec le réseau de communication. La station de transport réceptrice, effectue le traitement inverse avant de transférer les données à l'abonné destinataire. Ce traitement reste transparent, tant à l'émetteur qu'au récepteur.

Deux types de messages peuvent être échangés par les abonnés: Lettres et Télégrammes. Une Lettre est un message de taille variable destiné à l'échange des données. Un Télégramme est un message de taille fixe, de quelques octets, destiné au contrôle. Les Télégrammes sont envoyés à n'importe quel moment, sans tenir compte des lettres émises, reçues ou en cours d'émission, ou de réception.

En service de base, la station de transport assure seulement deux services : Multiplexage/Démultiplexage et Fragmentation/Ré-assemblage. Aucun contrôle n'est fait pour détecter ou récupérer des Lettres ou Télégrammes perdus, dupliqués ou reçus hors de séquence.

#### 1.3.4.1 Multiplexage/Demultiplexage

Les différents abonnés d'un ordinateur participant, ont accès au réseau (et aux abonnés distants) en passant par la station de transport locale. A un instant donné, chaque abonné peut avoir plusieurs portes actives, et plusieurs abonnés peuvent être actifs simultanément. La station de transport locale doit alors gérer plusieurs flots simultanément.

La station de transport assure le multiplexage des différents flots vers la ligne série qui relie l'ordinateur hôte avec un noeud du réseau de communication CIGALE. C'est en fait ce multiplexage qui rentabilise l'utilisation des lignes téléphoniques louées puisque leur coût est partagé entre tous les abonnés locaux.

En réception, la station de transport fait le démultiplexage en aiguillant chaque Lettre ou Télégramme reçu vers la porte appropriée de l'abonné local. Ce multiplexage/démultiplexage reste transparent aux abonnés, qui croient disposer de la ligne en permanence.

#### I.3.4.2 Fragmentation/Ré-assemblage

Le réseau de communication CIGALE, utilisant la technique de commutation de paquets, limite la taille maximale des paquets transportés. Pour cette raison, chaque fois qu'une Lettre émise par un abonné, dépasse la taille maximale (255 octets), la station de transport la découpe en fragments de taille appropriée et les transmet un par un au réseau de communication. Chacun de ces paquets est transporté par CIGALE indépendamment des autres.

Tous les fragments d'une Lettre ont la même taille excepté le dernier, ce qui facilite la tâche de ré-assemblage. Chaque fragment est envoyé avec une information de contrôle (numéro de Lettre, numéro de fragment) qui permet à la station de transport réceptrice de ré-assembler la Lettre avant de la délivrer au destinataire. La station de transport émettrice attribue une référence unique à chaque Lettre dans un flot. Le dernier fragment comporte une marque qui l'identifie et permet à la station réceptrice de connaître la longueur de la Lettre en fragments.

Le processus de ré-assemblage est protégé par une temporisation qui est ré-enclenchée à la réception de chaque fragment et finalement annulée dès que la Lettre est complète. Si la temporisation échoue, on abandonne les fragments reçus jusqu'à ce moment et on considère la Lettre comme perdue. La station de transport réceptrice ne délivre aux abonnés que les Lettres complètes reçues sans erreur.

Le processus de fragmentation/ré-assemblage reste transparent aux utilisateurs qui émettent ou reçoivent des Lettres complètes.

### 1.3.5 Services Additionnels

Des services additionnels peuvent être demandés en option à l'ouverture d'un flot. Ces services ne sont appliqués que si les deux correspondants sont d'accord pour les utiliser. Les services demandés restent actifs pendant toute la durée de la session. (On appelle session, le temps entre l'ouverture du flot et sa fermeture).

Trois types de services additionnels peuvent être demandés :

- contrôle d'erreurs,
- contrôle de flux,
- adressage réduit.

Un service additionnel est toujours actif dans les deux sens du dialogue.

#### 1.3.5.1 Contrôle d'Erreurs

Le service de contrôle d'erreurs sur un flot, a pour but de détecter et reprendre les erreurs dues à la duplication ou à la perte des lettres. Il assure également le bon séquençement des lettres. Ce service est exécuté en plus du service de base et il est effectif dans les deux sens du flot.

En émission, on donne à chaque lettre une référence séquentielle cyclique qui apparaît dans l'en-tête de chacun de ses fragments. A l'ouverture du flot, on initialise les références des lettres à zéro dans les deux stations de transport, émettrice et réceptrice. La première lettre émise par chaque station de transport sur ce flot, portera la référence 1.



La station de transport réceptrice acquitte les Lettres reçues sans erreur soit en envoyant une commande explicite d'acquiescement avec la référence de la Lettre bien reçue, soit en mettant cette référence dans le champ prévu à cet effet dans l'en-tête d'un fragment en émission sur le flot. En réception, les Lettres dupliquées (avec la même référence) sont abandonnées et ne sont pas transférées aux abonnés.

La réception d'un acquiescement est contrôlée par une temporisation, après laquelle la lettre est considérée comme perdue et renvoyée depuis le premier fragment. Si une Lettre a été émise N fois sans succès, la station de transport émettrice déclare une erreur irrécupérable et ferme le flot. La réception d'un acquiescement avec la référence "i" implique que la Lettre de référence "i" et toutes les précédentes ont été bien reçues.

Pour maintenir un trafic minimal sur un flot, les stations de transport envoient un message d'acquiescement tous les intervalles de temps T01, avec la référence de la dernière Lettre bien reçue ou, si aucune n'a été bien reçue, avec la référence zéro.

#### 1.3.5.2 Contrôle de Flux

Le service de contrôle de flux est exécuté en plus du service de contrôle d'erreurs. Il a pour but de régler le débit des échanges sur un flot en fonction des possibilités de stockage et de traitement de chacun des partenaires.

A l'ouverture du flot, les abonnés doivent spécifier la taille maximale des Lettres à échanger. Cette taille peut être différente pour chacun des abonnés.

Le contrôle de flux est implémenté à l'aide d'un système de crédits envoyés par chacune des stations de transport. Une unité de crédit indique que l'abonné peut recevoir une Lettre de taille égale ou inférieure à la taille maximale spécifiée à l'ouverture du flot.

La station de transport émettrice, n'envoie une Lettre que si elle a reçu un crédit de la part de la station de transport réceptrice, indiquant que son abonné peut la recevoir.

La station de transport réceptrice envoie à la station émettrice des crédits au fur et à mesure que son abonné lui indique sa capacité de stockage ou de traitement. Si un message de crédit se perd, il existe la possibilité d'un blocage du flux sur un flot, car la station de transport émettrice attend ce message de crédit avant d'émettre. Pour prévenir cette éventualité, la station de transport réceptrice envoie un message de crédit tous les intervalles de temps T02.

#### 1.3.5.3 Adressage Réduit

L'option d'adressage réduit permet l'identification d'un flot non plus par la paire de ses adresses des portes, mais à l'aide d'un identificateur propre appelé "plug". Ceci a pour but d'alléger la tâche d'identification d'un flot à la réception de chaque message.

Cette option n'a pas encore été implémentée sur le réseau CYCLADES.



CHAPITRE IIIMPLEMENTATION DE LA LA STATION DE TRANSPORT

## 1 - Introduction

## 2 - Implantation de la Station de Transport

- .1 Intégration au système d'exploitation
- .2 Implantation en mode usager
- .3 Implantation dans un ordinateur frontal

## 3 - Spécifications de réalisation de la ST

- .1 Les interfaces externes
- .2 Les Processus de la ST
  - .1 Le processus P-IAB
  - .2 Le processus P-EMIS
  - .3 Le processus P-RECEP
  - .4 Le processus P-REVEIL
  - .5 Le processus P-RESEAU
  - .6 Le Sélecteur

## 4 - Implémentation de la ST sur l'IBM 360/67

- .1 Le système SYNCOP
- .2 La Station de Transport ,
  - .1 Le processus LIGNE
  - .2 Le processus RECEPTION
  - .3 Le processus EMISSION
  - .4 Le processus REVEIL
- .3 Interface avec les abonnés
  - .1 Interface abonnés internes
  - .2 Interface abonnés externes



## II.1 - INTRODUCTION

La station de transport est une couche logicelle d'interface entre les utilisateurs locaux d'une installation et le réseau d'ordinateurs. Elle peut être vue comme une méthode d'accès au réseau qui fournit aux utilisateurs un ensemble de fonctions de haut niveau leur permettant de dialoguer avec des correspondants distants.

Chaque ordinateur participant doit posséder obligatoirement une station de transport qui offre ses services aux abonnés locaux. Les stations de transport dialoguent entre elles conformément au protocole de transport. La communication est assurée par le réseau de communication CIGALE.

L'ensemble des stations de transport implantées sur les ordinateurs participants, constituent la machine de transport du réseau.

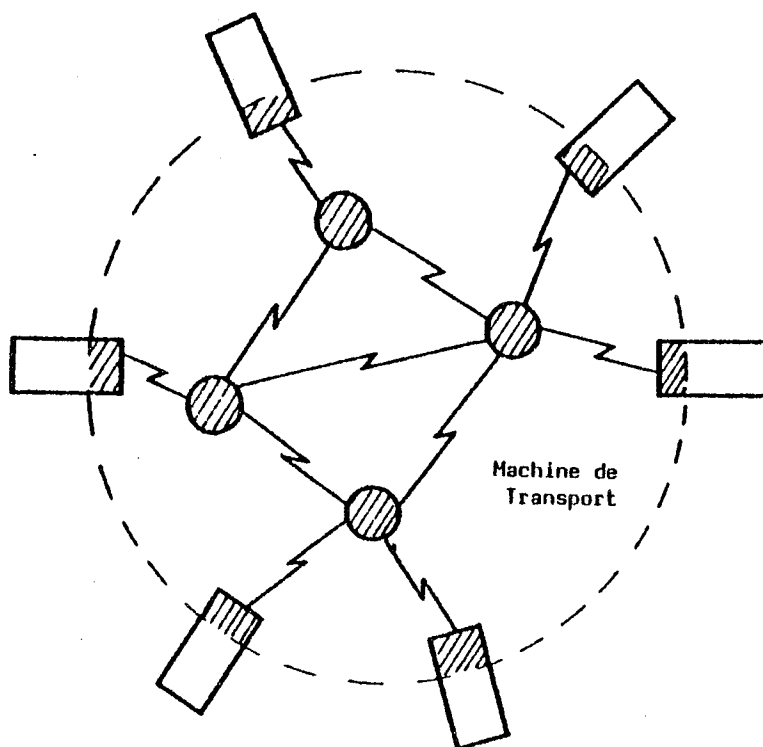


Fig. II-1 La Machine de Transport

## II.2 - IMPLANTATION DE LA STATION DE TRANSPORT

Nous avons vu précédemment que la connexion d'un ordinateur au réseau CYCLADES, impose l'implantation d'une station de transport d'interface avec le réseau. La mise en oeuvre de la station de transport constitue un des plus grands obstacles pour relier un ordinateur au réseau. plus grands obstacles pour relier un ordinateur au réseau.

Il existe plusieurs méthodes d'implantation de la station de transport sur un ordinateur (17). Parmi lesquelles on peut mentionner :

- intégration au système d'exploitation,
- implantation en mode usager,
- implantation dans un ordinateur frontal.

### II.2.1 Intégration au système d'exploitation

Une méthode très commode pour implanter une station de transport sur un ordinateur est de l'intégrer au système d'exploitation du site. De cette façon, la station de transport se présente comme les autres méthodes d'accès du système avec une interface semblable. Les abonnés locaux peuvent s'adresser aux ressources du réseau de manière transparente, comme s'il s'agissait d'une ressource locale.

Dans ce cas la Station de Transport inclut, en plus des fonctions définies pour le protocole de transport, la gestion de la ligne synchrone qui relie l'ordinateur à un noeud de CIGALE.

Cette solution serait idéale si l'intégration était prévue ou faite dans le logiciel système du constructeur de l'ordinateur. Malheureusement, ceci n'est généralement pas le cas, tout du moins jusqu'à présent.

L'intégration de la station de transport au système est une tâche difficilement abordable qui nécessite un effort considérable d'écriture de programmes, de modification du système et de mise au point. Or, dans la plupart des cas, les utilisateurs ne disposent pas d'une équipe système assez forte, ou ne disposent pas des moyens nécessaires (en hommes/années) pour aborder un tel travail. En plus, dès que l'on s'attaque à modifier le système d'exploitation, on diminue souvent sa fiabilité en risquant d'occasionner des "cassures de système" assez fréquentes, ce qui entraîne une dégradation des services offerts par l'installation qui dans la plupart des cas est inacceptable.

D'autre part, le fait que le logiciel de connexion au réseau soit exécuté par l'ordinateur hôte diminue forcément ses performances et introduit des contraintes de temps dues à la gestion de la ligne synchrone et aux temporisations associées aux différents protocoles.

### II.2.2 Implantation en mode Usager

La méthode la plus courante pour implanter une station de transport, est de la présenter au système d'exploitation comme un usager du système.

La liaison avec le réseau est faite grâce à une ligne de transmission qui lui est affectée en permanence. La station de transport ne gère pas la ligne directement mais utilise la méthode d'accès tele-informatique "standard" de l'installation.

Les programmes abonnés sont situés dans d'autres tâches du système. Les échanges entre la station de transport et les abonnés doivent se faire à l'aide des mécanismes de communication inter-tâches prévus dans le système d'exploitation (boîte aux lettres, files d'attente, interconnexion de canaux entre machines virtuelles, etc.).



Fréquemment on inclut aussi dans la tâche de la station de transport, les protocoles d'application les plus courants (protocole d'appareil virtuel, protocole de soumission des travaux à distance, etc.).

La principal avantage de cette méthode d'implantation de la station de transport, réside dans le fait qu'elle ne nécessite aucune modification au système d'exploitation du site puisque simplement il suffit de définir une nouvelle tâche, qui est active tant que l'on désire maintenir la liaison avec le réseau.

L'effort (en hommes/années) nécessaire à cette implantation est nettement inférieur à celui de la méthode précédente et n'exige pas une si bonne connaissance du système d'exploitation. La mise au point peut être faite en parallèle des autres activités du centre de calcul, sans gêner le fonctionnement normal du système.

Le principal inconvénient de cette méthode d'implémentation est que les abonnés doivent s'adresser au réseau au moyen d'une méthode d'accès particulière. On ne peut donc pas banaliser les ressources du site avec celles du réseau.

D'autre part il existe le même inconvénient que dans le cas précédent en ce qui concerne la diminution des performances du système et les contraintes de temps réel introduites.

Malgré ces inconvénients, les avantages sont très nombreux, ce qui explique pourquoi on a implanté la station de transport par cette méthode dans la plupart des ordinateurs reliés au réseau CYCLADES.

Afin de faciliter le raccordement d'un ordinateur au réseau, une version de la station de transport a été écrite dans un langage portable (19). Ceci simplifie la tâche de son implantation en limitant les efforts nécessaires, pour l'équipe

système du site, à l'écriture d'un interpréteur du langage portable ou à la translation de ce langage dans le langage système de cette machine.

### II.2.3 Implantation dans un Ordinateur Frontal

Un autre méthode de raccordement d'un ordinateur au réseau consiste à utiliser un autre ordinateur dit "frontal" comme intermédiaire. Ce frontal, généralement un mini-ordinateur, se présente à l'ordinateur hôte comme un ensemble de ressources (18). Vers le réseau, le frontal se présente comme étant l'ordinateur hôte.

Les terminaux ou ressources simulés par le frontal vis à vis de l'ordinateur, sont adressés par les utilisateurs locaux suivant les conventions de la méthode d'accès aux ressources externes simulables du système d'exploitation. Le frontal fait la conversion de ces conventions locales aux conventions du réseau.

Cette méthode regroupe les avantages des deux méthodes mentionnées précédemment. En effet, les utilisateurs locaux peuvent s'adresser aux ressources du réseau en utilisant les méthodes d'accès définies par le constructeur, et il n'est donc pas nécessaire de modifier le système d'exploitation.

Cependant, cette méthode présente un certain nombre de désavantages qui expliquent pourquoi elle n'est pas employée plus fréquemment. Parmi ceux-ci, on peut mentionner :

#### 1) Le raccordement du frontal à l'ordinateur hôte.

Ce raccordement doit être simple tout en ayant un débit de transfert qui ne limite pas ses possibilités. En général, cette liaison est faite par une ligne série rapide connecté à un contrôleur de communication de téléinformatique de l'ordinateur ce qui lui maintient le même genre de problèmes à résoudre.

## 2) L'écriture du logiciel frontal

Généralement, on achète l'ordinateur frontal comme une machine "nue", c'est à dire sans aucun système d'exploitation . Les systèmes d'exploitation commercialisés couramment étant généralement mal adaptés à cette application. Il reste donc à l'équipe système locale l'écriture de ce logiciel d'interface vers le réseau et d'interface vers l'ordinateur hôte.

## 3) Le coût du frontal

L'ordinateur frontal est généralement un mini-ordinateur dont le coût est assez élevé si l'on ne s'en sert que pour faire l'interface avec le réseau. Une alternative est de l'utiliser également pour d'autres applications telles que : concentrateur de terminaux (18), entrée des données en temps réel etc. Dans ce cas le frontal devient plus rentable, mais ceci dépend seulement des besoins du site. Si l'on desire seulement se raccorder au réseau, cette solution n'est pas rentable.

### II.3 - SPECIFICATIONS DE REALISATION DE LA ST

Afin de faciliter l'implantation de la station de transport en mode usager dans différents modèles et marques d'ordinateurs utilisant différents systèmes d'exploitation, une version "portable" de la station de transport ST2 a été écrite.

Dans cette version, la station de transport se présente au système d'exploitation comme un usager du système. Jusqu'à présent, elle a été implémentée dans la plupart des ordinateurs reliés à CYCLADES.

Les spécifications de réalisation de cette station de transport définissent la décomposition du logiciel de transport en processus ainsi que les mécanismes de communication et synchronisation entre processus (19).

#### II.3.1 Les Interfaces Externes

La station de transport communique d'un côté avec les abonnés locaux en passant par le système d'exploitation, et de l'autre côté avec le réseau via une ligne de communication série.

Les interfaces sont faits grâce à un mécanisme de files. La station de transport reçoit des blocs de demande (BD) de l'extérieur et une fois que la requête a été traitée, elle répond par un bloc de réponse (BR).

L'interface avec les abonnés est fait à l'aide des "canaux". Un canal se compose d'une file de blocs de demande (BD) de l'abonné vers la ST et d'une file de blocs de réponse (BR) de la ST vers l'abonné.

Pour chaque flot il existe trois canaux :

- un canal pour la réception de Lettres,
- un canal pour l'émission de Lettres,
- un canal pour l'émission/réception de télégrammes.

En plus il existe un canal commun à tous les abonnés pour les ouvertures/fermetures des flots.

L'interface avec le réseau se fait à l'aide de deux canaux uniques. Un pour l'émission de messages, et l'autre pour la réception. Ces canaux relient la station de transport au gérant de la ligne qui effectue la liaison de l'ordinateur avec un noeud du réseau.

### II.3.2 Les Processus de la ST

Dans les spécifications de réalisation de la station de transport mentionnées précédemment, le logiciel de transport est divisé en 5 processus :

- Processus Interface Abonné P-IAB,
- Processus d'Emission P-EMIS,
- Processus de Reception P-RECEP,
- Processus de Reveil P-REVEIL,
- Processus Réseau P-RESEAU.

#### 3.2.1 Le Processus Interface Abonné

Le processus Interface Abonné, tel que son nom l'indique, effectue l'interface entre la station de transport et les abonnés du site. Les communications sont faites à l'aide des canaux décrits précédemment.

Le processus P-IAB, reçoit les blocs de demande (BD) qui contiennent les requêtes des abonnés, les traite et les réfléchit vers le processus d'émission P-EMIS.

Une fois qu'une requête a été traitée, P-IAB (ou P-EMIS) envoie à l'abonné correspondant, un bloc de réponse (BR) en le lui indiquant.

Le processus P-IAB gère les requêtes des abonnés à l'aide d'une table d'événements. A chaque liste de blocs de demande est associé un événement permettant l'activation du processus P-IAB quand un bloc de demande est déposé dans la liste par un abonné. En conséquence, P-IAB ne traite sur une liste qu'une demande à la fois.

#### II.3.2.2 Le Processus d'Emission

Le processus d'émission P-EMIS prépare les paquets à transmettre vers le réseau. Il balaie la table de flots actifs (TFLACTIF), en sélectionne un parmi ceux qui sont prêts à émettre et transmet le paquet correspondant vers le processus P-RESEAU. Une fois que ce dernier a pris en compte le paquet, il prévient P-EMIS, autorisant celui-ci à transférer un nouveau paquet vers P-RESEAU.

#### II.3.2.3 Le Processus de Réception

Le processus de réception P-RECEP accepte les paquets que lui transfère P-RESEAU qui ont été reçus du réseau. Il cherche le flot concerné et met à jour la table du contexte du flot. Si la Lettre est complète (ré-assemblage fini), il prévient l'abonné destinataire. Si des services additionnels sont exécutés sur le flot, il construit le message d'accusé de réception et met à jour la table de flots (TFLOTS).

#### II.3.2.4 Le Processus de Réveil

Le processus de réveil P-REVEIL tient à jour l'heure locale de la station de transport et traite toutes les temporisations concernant :

- le ré-assemblage des Lettres,
- les réceptions des accusés de réception,
- les réponses aux ouvertures de flot,
- le trafic sur chaque flot.

Ce processus est activé cycliquement à chaque quantum de temps. Il fait la mise à jour de l'heure locale et vérifie s'il y a une ou plusieurs temporisations qui ont échoué.

#### II.3.2.5 Le Processus Réseau

Le processus réseau P-RESEAU, assure l'interface de la station de transport vers le réseau en dialoguant avec le gérant de la ligne. Ce dialogue se fait à l'aide des canaux décrits précédemment.

Il est activé en émission par P-EMIS et en réception par le gérant de la ligne.

#### II.3.2.6 Le Sélecteur

Le sélecteur est une procédure commune aux processus P-IAB, P-EMIS et P-RECEP. Son rôle est de remplir la table de flots actifs TFLACTIF en balayant la table générale des flots TFLOTS, et en choisissant ceux qui ont des paquets à émettre. En quelque sorte il est un allocateur des fonctions de la ST aux flots.

Le sélecteur est appelé par P-EMIS après chaque émission de paquet, par P-IAB chaque fois qu'il traite une demande d'un abonné, et par P-RECEP lorsque celui-ci doit envoyer un message d'acquittement.

Cette procédure joue un rôle prépondérant dans cette décomposition de la station de transport. Elle gère les tables TFLOTS et TFLACTIF qui sont utilisées par les processus pour communiquer et se synchroniser.

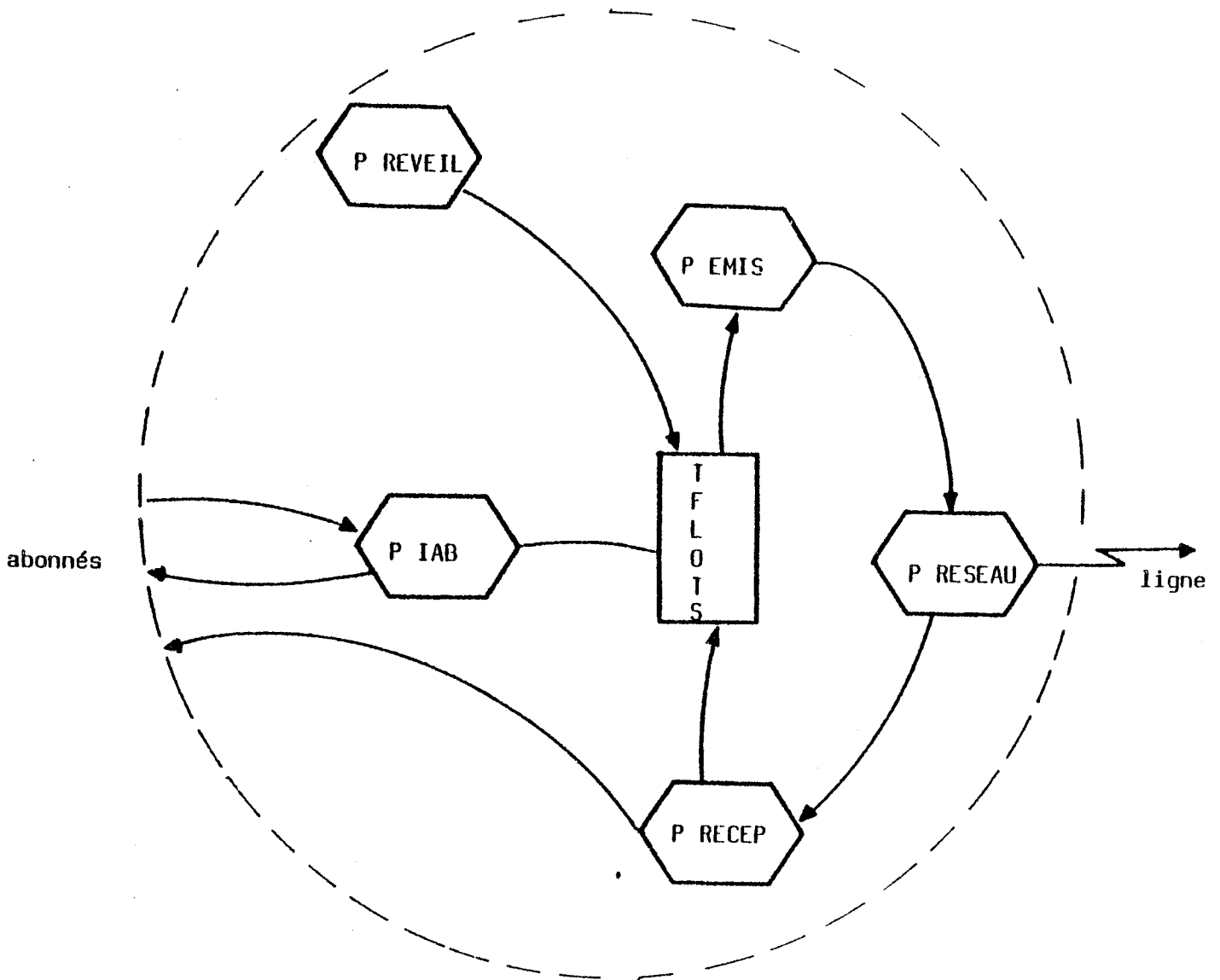


Fig. II - 2 La Structure de la ST2



## II.4 - IMPLEMENTATION DE LA ST SUR L'IBM 360/67

L'ordinateur IBM 360/67 du Centre Inter-Universitaire de Calcul de Grenoble tourne sous un système d'exploitation particulier appelé CP/67. ce système offre à chaque utilisateur une machine virtuelle qui est une simulation logicielle d'une machine IBM 360 réelle et de ses périphériques associés.

La configuration de chaque machine virtuelle peut varier selon les demandes et besoins de l'utilisateur. Comme les machines réelles, les machines virtuelles fonctionnent sous un système d'exploitation. Ce système peut être différent pour chaque machine virtuelle, tel que : un système propre à IBM comme CMS, OS/360, DOS, APL/360, CP/67 lui même; ou un système défini par l'utilisateur comme TELCOM (20) ou SYNCOP (21).

La station de transport du réseau CYCLADES est implémentée sur une machine virtuelle simulée par CP/67. Cette machine tourne sous le système d'exploitation SYNCOP, conçu spécialement pour des applications de télétraitement. Elle a, attachée en permanence, une ligne série pour la connexion avec un noeud du réseau de communication, une console opérateur et l'espace disque et mémoire nécessaire pour son fonctionnement.

### II.4.1 Le Système d'exploitation SYNCOP

SYNCOP est un système de communication de processus qui présente une adaptabilité maximale aux applications télé-informatiques (22). Dans sa version originale il était prévu pour tourner comme un sous-système sous le système hôte, en se présentant comme une tâche utilisateur. Dans le cas de CP/67, il tourne comme un système d'exploitation sur une machine virtuelle.

Les services offerts par SYNCOP peuvent être résumés dans la liste suivante :

- gestion de processus,
- synchronisation entre processus,
- gestion de mémoire,
- gestion du temps,
- gestion de files,
- gestion de ressources,
- gestion des entrées/sorties.

SYNCOP est un système non préemptible. Quand une interruption se produit, le processus actif perd le contrôle de l'unité centrale au profit du superviseur qui traite l'interruption. Une fois l'interruption traitée, le commutateur rend le contrôle au processus interrompu.

#### II.4.2 La Station de Transport

L'implémentation de la station de transport du réseau CYCLADES sur une machine virtuelle est inspirée de la description faite pour la version portable de la SI2 décrite précédemment. Cependant, certaines modifications ont été faites dans les mécanismes de communication et de synchronisation entre processus et avec les abonnés pour les adapter à l'environnement de CP/67 et du système d'exploitation SYNCOP.

Sur la machine virtuelle, la station de transport est composée principalement de 3 processus : LIGNE, EMISSION, et RECEPTION (8). Un quatrième processus REVEIL intervient pour les temporisations, mais en réalité il appartient au système SYNCOP lui même.

Ces processus sont les équivalents des processus P-RESEAU, P-EMIS et P-REVEIL mentionnées précédemment dans la description de la station de transport portable.

#### II.4.2.1 Le processus LIGNE

Le processus LIGNE fait l'interface entre la station de transport et le réseau. Il gère la procédure de ligne de liaison de l'IBM 360 avec le noeud de CIGALE. La connexion est faite sur une ligne synchrone en mode "half-duplex" du contrôleur de communication IBM 2701 et gérée par la procédure BSC.

Le processus LIGNE puise ses informations dans une liste d'émission ENVOIPQ dans laquelle le processus EMISSION a produit des paquets destinés à être envoyés sur le réseau. Par ailleurs, il consomme les paquets qui lui arrivent de la ligne et les range dans la liste de réception RECAPPQ.

LIGNE ne range dans RECAPPQ que les paquets reçus sans erreur et envoie vers le noeud les messages d'acquiescement prévus par la procédure. D'autre part, il retransmet les paquets qui n'ont pas été acquiescés par le noeud dans le délai prévu.

#### II.4.2.2 Le Processus RECEPTION

Le processus RECEPTION traite les paquets reçus dans la liste RECAPPQ et ne se met en attente que si cette liste est vide. Il analyse les paquets provenant du réseau et exécute un traitement différent pour chaque type de message. Trois types de messages sont prévus :

- Télégrammes,
- fragments de Lettres,
- commandes de négociation.

RECEPTION abandonne les paquets qu'il ne peut pas traiter à cause d'une mauvaise entête de paquet ou de commande, les paquets destinés à un abonné non existant, et les paquets sans signification.

Les télégrammes ne sont livrés aux abonnés destinataires que si ceux-ci ont fait une demande explicite de les recevoir.

RECEPTION ré-assemble les fragments d'une lettre et, si elle est complète, la délivre à l'abonné destinataire en lui indiquant de plus sa provenance. Ce dernier point est important car il est possible qu'un abonné reçoive des paquets sur une porte sans qu'il existe un flot établi.

Si la lettre est reçue sur un flot avec services additionnels, RECEPTION génère l'acquittement correspondant. En plus, il vérifie le séquençement des lettres reçues sur un même flot et ne délivre une lettre de référence "i" que si celle de référence "i-1" a été bien reçue et délivrée.

RECEPTION traite les commandes de négociation suivant le protocole de négociation. Il annule les temporisations associées à leur réception et génère les messages de réponse prévus par le protocole. Dès que le flot est ouvert, il informe l'abonné correspondant.

#### II.4.2.3 Le Processus EMISSION

Le processus EMISSION prépare les paquets à envoyer sur le réseau via le processus LIGNE. Un paquet peut être soit: un télégramme, un fragment de Lettre ou une commande de négociation. Dans chaque cas, EMISSION prépare l'entête et range le paquet dans la liste ENVOIPQ pour le processus LIGNE. Si cette liste est pleine, et qu'il y ait toujours des paquets à émettre, EMISSION se met en attente de l'événement "liste ENVOIPQ non pleine".

Si le paquet à émettre est destiné à un abonné local, dans la même station de transport, EMISSION le range directement dans la liste RECPPQ du processus RECEPTION qui le traite comme si on l'avait reçu du réseau (routage interne).

Si la taille d'une Lettre dépasse la taille maximale définie dans le protocole de négociation à l'ouverture du flot correspondant, le processus EMISSION la découpe en fragments égaux (sauf le dernier). EMISSION ajoute à chaque fragment l'information de contrôle nécessaire pour que la Lettre puisse être ré-assemblée à la réception.

Si le flot est soumis au contrôle d'erreurs, EMISSION enclenche une temporisation d'attente de l'acquittement correspondant.

Si le flot est soumis au contrôle de flux, EMISSION vérifie que l'abonné émetteur dispose de "crédits" chez son partenaire récepteur.

A l'émission d'une commande de négociation, EMISSION enclenche la temporisation d'attente de réception de la commande correspondante prévue dans le protocole de négociation.

#### II.4.2.4 Le Processus REVEIL

Le système SYNCOP prévoit deux types de temporisations pour les processus qu'il gère :

- Un processus veut attendre pendant une durée déterminée avant de reprendre l'exécution. A ce type de réveil est associé un événement qui sera réalisé quand le temps donné sera écoulé.

- Un processus souhaite qu'au bout d'un temps donné, une procédure déterminée soit exécutée. Ce réveil avec procédure associé est utilisé par la station de transport pour réaliser le traitement de contrôle d'une horloge de garde.

Dans le logiciel de la station de transport, quatre types de traitements sont prévus :

- temporisation de garde de ré-assemblage d'une Lettre,
- temporisation de garde d'attente de réception d'une commande de négociation,
- temporisation de garde d'attente d'acquittement de réception d'une Lettre sous contrôle d'erreurs,
- temporisation d'entretien d'activité minimale sur un flot.

Les procédures associées à ces temporisations se déroulent sous contrôle du processus REVEIL de SYNCOP.

A chaque temporisation correspond un bloc réveil BRV spécifiant la durée de la temporisation, l'adresse de la procédure associée, l'identification du processus qui a lancé l'horloge de garde et la référence de la Lettre et/ou flots concernées.

Quand une interruption externe de type horloge se produit, le processus actif perd le contrôle de l'unité centrale au profit du superviseur des interruptions externes, EXTINT. Celui-ci place le bloc réveil associé à la temporisation qui a échoué dans la pile du processus REVEIL, met ce dernier à l'état "activable" et rend la main au processus interrompu. Quand celui-ci se mettra en attente, le processus REVEIL sera activé avant les autres car il est de priorité d'activation plus élevé. REVEIL à son tour, exécutera la procédure associée à la temporisation qui a échoué.

### II.4.3 Interface avec les Abonnés

Les machines virtuelles simulées par le système CP/67 ont une taille de mémoire minimale de 256 Koctets. La station de transport et le système SYNCOP n'occupent qu'environ un tiers de la mémoire de la machine virtuelle sur laquelle ils sont implantés. Pour cette raison, on a également inclus dans cette machine virtuelle certains processus des abonnés spécialisés.

Ces abonnés sont des abonnés système, de type serveur ou concentrateur, qui permettent l'accès des abonnés distants aux services offerts par le système CP/67.

De ce fait, deux types d'abonnés existent : abonnés internes et abonnés externes.

#### II.4.3.1 Interface Abonné Interne

Les abonnés internes sont les abonnés système qui se trouvent dans la même machine virtuelle que la station de transport s'exécutant sous le système d'exploitation SYNCOP.

La coexistence des abonnés internes et la station de transport dans le même espace de travail rend caduque et inefficace l'existence d'un processus d'interface abonné-station de transport.

La communication entre les abonnés internes et la station de transport est faite à l'aide d'un ensemble de procédures ré-entrantes, spécialisées dans la communication et mises au service des abonnés.

Ces procédures permettent aux abonnés de réaliser les demandes suivantes :

- création et fermeture d'une porte,
- création et fermeture d'un flot,
- envoi et réception d'une Lettre,
- envoi et réception d'un télégramme.

La communication de la station de transport vers les abonnés internes, s'effectue par appel d'une procédure unique spécialisée REPAB qui produit un bloc de réponse BR dans la liste prévue par l'abonné.

#### II.4.3.2 Interface Abonné Externe

Les abonnés externes sont ceux qui s'exécutent dans d'autres machines virtuelles simulées par CP/67 que celle de la station de transport.

On distingue deux types de machines virtuelles abonnés :

- Machines virtuelles fonctionnant sous le système SYNCOP dans laquelle il peuvent y avoir plusieurs processus abonnés différents concurrençant à l'accès aux services partagés de la station de transport.
- Machines virtuelles fonctionnant sous le système CMS dont un seul processus dispose du contrôle de l'unité centrale et de la mémoire.

Dans les deux cas, l'interface externe avec les abonnés et la station de transport est faite par un processus INTAB, s'exécutant sur la machine virtuelle de la station de transport qui gère les liens de communication (boîte aux lettres, files SYNCOP, files).



Les abonnés produisent dans le lien des blocs de demande BD contenant leurs requêtes. INTAB lit ces blocs de demande et, selon le type, fait appel à la procédure de traitement correspondante. Les blocs de réponse BR, transitent par ce lien selon le même principe.

Le lien de communication entre les abonnés et la station de transport est réalisé à l'aide de l'outil logiciel de communication inter-machines virtuelles disponible sur le système CP/67. Au moyen de cet outil, appelé BALCP, la communication est faite selon le principe d'une boîte aux lettres.

Un couple de processus (BALSTE, BALSTL) par liaison inter-machines virtuelles gère la communication. Dans le cas d'une machine virtuelle fonctionnant sous SYNCOP, les demandes et les réponses de plusieurs abonnés sont multiplexées sur le même lien.

BALSTL démultiplexe les blocs de demande BD de l'abonné externe et appelle les procédures internes de l'interface des abonnés internes. BALSTE multiplexe les blocs de réponse BR et les expédie vers l'abonné. Un processus BALIN est à l'écoute des demandes d'ouverture de liaisons inter-machines virtuelles. A chacune des ces demandes, il crée une paire de processus BALSTL, BALSTE.

Il faut noter que ces liens sont très différents de ceux qui sont utilisés pour l'utilisation des ressources du système.

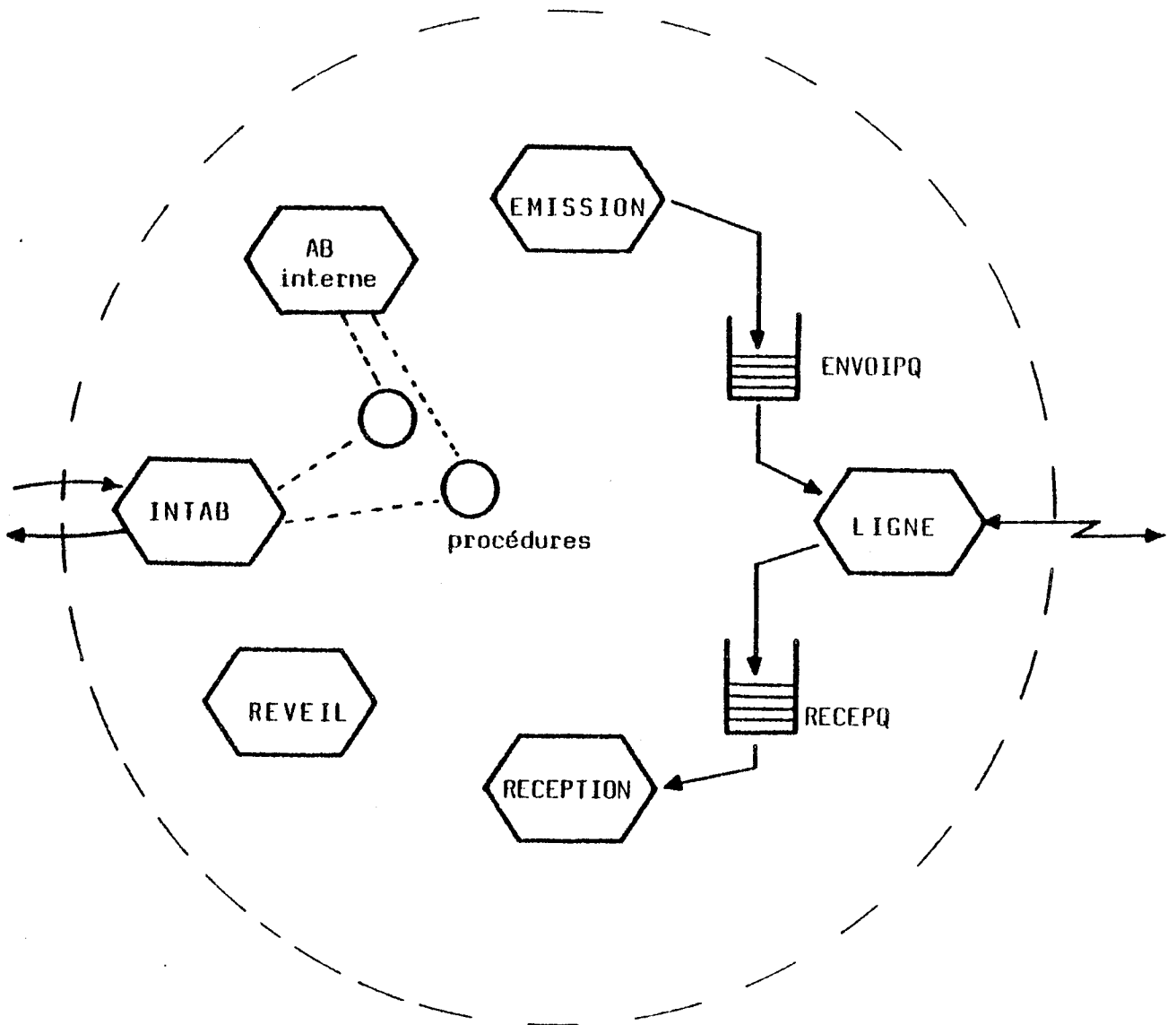


Fig. II-3 Structure de la ST sous CP/67



CHAPITRE IIIDECOMPOSITION DE LA STATION DE TRANSPORT

## 1 - Introduction

## 2 - Décomposition de la ST

- .1 Interface avec les abonnés
- .2 Fragmentation/ré-assemblage
- .3 Division des fonctions

## 3 - La Station Résiduelle

- .1 Description de fonctions
- .2 Implémentation de la Station Résiduelle
  - .1 Implémentation en mode usager
  - .2 Implémentation intégrée au système

## 4 - La Station Externe

- .1 Description fonctionnelle
  - .1 Négociation de session
  - .2 Fragmentation/ré-assemblage
  - .3 Contrôle d'erreurs
  - .4 Contrôle de flux
  - .5 Temporisations
  - .6 Télégrammes
- .2 Interface avec la Station Résiduelle
- .3 Implémentation de la Station Externe

## 5 - Définition du Système PIAR

- .1 Description fonctionnelle
  - .1 Station de Transport
  - .2 Liaison avec le réseau
  - .3 Liaison avec l'ordinateur hôte
- .2 Considérations de compatibilité
- .3 Considérations de débit

### III.1 - INTRODUCTION

Tel que nous l'avons mentionné dans le chapitre précédent, la station de transport servant d'intermédiaire entre le réseau CYCLADES et l'ordinateur IBM 360/67 du CICG est implémentée sur une machine virtuelle simulée par CP/67. En quelque sorte on peut voir cette implémentation comme étant en mode frontal si l'on considère que la machine virtuelle est un ordinateur frontal virtuel placé entre le réseau et les autres machines virtuelles. D'autre part, la machine virtuelle est un usager du système d'exploitation CP/67, et elle s'exécute dans le même environnement que ses utilisateurs. On peut donc aussi considérer que la ST est implémentée en mode usager.

Cette implémentation comporte un certain nombre d'inconvénients d'Os, soit au système d'exploitation, soit à la configuration du site. Parmi ces inconvénients on peut mentionner les suivants :

#### 1) Le système d'exploitation

Le système d'exploitation CP/67 est mal adapté aux applications de télé-traitement. Il simule des machines virtuelles, parmi lesquelles celle de la station de transport, sans distinction les unes des autres. Le système de temps partagé ne tient pas compte du fait qu'une machine a plus de charge ou plus d'événements en temps réel à gérer que les autres pour lui allouer plus ou moins de quanta de temps.

Une machine virtuelle n'est pas maîtresse du contrôle du temps de l'unité centrale et, à priori elle ne sait pas à quel moment le système de temps partagé va lui allouer un, ou plusieurs, quanta de temps d'unité centrale. De même, elle ne sait pas quand elle va perdre le contrôle de celle-ci.

Lorsque la charge du système augmente, le nombre et la fréquence des quanta qui sont alloués à une machine virtuelle diminue. Sur la machine virtuelle ST, il arrive alors le phénomène bien connu de saturation du système quand elle ne dispose pas d'assez de quanta d'unité centrale pour son autogestion et pour traiter toutes les requêtes des abonnés locaux ou distants.

Le fait qu'une machine virtuelle sature n'implique pas que les autres machines virtuelles ni le système CP/67 lui même, aient le même problème. Ceci parce que CP/67 traite toutes les machines virtuelles indistinctement, sans priorité, et il n'y a aucun moyen sous CP/67 d'assurer le temps d'unité centrale dont une machine saturée a besoin au moment où elle en a besoin.

## 2) La configuration du site

La connexion entre l'ordinateur IBM 360 et un noeud de CIGALE est faite par une ligne synchrone attachée au contrôleur de communication IBM 2701. Ce contrôleur ne "connait" que la procédure de communication synchrone BSC. Il a donc fallu, pour faire cette connexion, développer un logiciel spécial sur le noeud CIGALE comportant cette procédure. En plus, BSC travaille en mode "half duplex" au lieu du mode "full duplex" de la procédure TMM-UCIG. On aurait pu utiliser deux lignes en mode "half-duplex" pour simuler une communication "full-duplex" mais on ne disposait plus de lignes libres sur le contrôleur.

En conséquence, le noeud de CIGALE auquel est relié l'IBM 360 est une exception dans le réseau car il doit gérer en plus des autres, une ligne avec la procédure BSC.

Ces raisons nous ont amenés à l'étude d'une autre méthode pour implémenter la station de transport de façon à éviter les inconvénients que l'on vient de mentionner. Nous avons d'abord écarté la possibilité d'utiliser un contrôleur de communication

évolué de la série IBM 38xx en raison de son coût élevé. Bien que ces contrôleurs soient très performants, ils réalisent bien plus de fonctions que celles que l'on désire et l'on se retrouve dans le cas d'implantation de la station de transport sur un mini-ordinateur frontal.



### III.2 - DECOMPOSITION DE LA ST

D'une étude préliminaire sur l'implémentation de la ST et de la fonction de communication avec CIGALE sur un frontal constitué d'une grappe de microprocesseurs (23), nous avons déduit qu'il était préférable de ne pas implémenter toutes les fonctions et parties de la ST sur ce frontal.

En particulier nous avons trouvé pratique de faire exécuter par l'ordinateur hôte, l'interface avec les abonnés et les tâches de fragmentation/ ré-assemblage.

#### III.2.1 Interface avec les abonnés

Tous les abonnés locaux de la ST sont, soit des utilisateurs, soit des abonnés-système type serveur ou concentrateur, qui s'exécutent sous le système d'exploitation CP/67. Dans les deux cas, ces abonnés sont des utilisateurs du système d'exploitation CP/67, de même que la partie de la station de transport s'exécutant sur l'ordinateur IBM 360.

Il nous a paru évident qu'il était plus facile et plus simple que la fonction d'interface entre les abonnés et la ST soit faite sous le système d'exploitation du site, en utilisant les liens et les primitives de communication qu'il fournit. Dans le cas de CP/67, ce sont les boîtes aux lettres (BALCP), inter-machines virtuelles; et dans le cas de SYNCOP, ses propres files. Dans les deux cas, ces communications ne représentent que des transferts de mémoire à mémoire dans la mémoire centrale de l'ordinateur hôte.

### III.2.2 Fragmentation/ré-assemblage

Le choix de laisser les tâches de fragmentation/ré-assemblage dans l'ordinateur hôte est dû surtout à l'importance de la taille des tampons de mémoire requis par ces tâches.

En principe, dès qu'un abonné manifeste son désir d'envoyer ou de recevoir une lettre, c'est à lui de fournir le tampon de mémoire (vide ou plein) correspondant. En émission, ce tampon ne peut être libéré tant que tous les fragments de la Lettre n'ont pas été envoyés, et que l'acquiescement correspondant n'a pas été reçu (cas de contrôle d'erreurs). En réception, un tampon est rendu à l'abonné seulement quand tous les fragments de la Lettre correspondante ont été bien reçus. Si on exécute en plus le service de contrôle d'erreur, alors on ne rend pas le tampon avec la Lettre de référence "i" que si celle avec la référence "i-1" a été déjà délivrée.

Ceci implique qu'à un instant donné il y a au moins deux tampons de mémoire immobilisés par flot. Ce nombre doit être multiplié par le facteur d'anticipation correspondant dans le cas où l'on utilise le service de contrôle de flux sur ce flot.

Si on voulait exécuter les fonctions de fragmentation/ré-assemblage sur le système frontal, il faudrait, deux conditions.

D'une part, que celui-ci dispose d'une zone de mémoire importante pour contenir les lettres en cours d'émission et de réception des différents flots ouverts.

D'autre part, en réception, il faudrait de toutes façons que l'abonné fournisse les tampons de mémoire pour toutes les Lettres qu'il veut recevoir, car il ne sait pas a priori quand il va les recevoir. Il y aurait donc un gaspillage de mémoire puisque l'on immobiliserait deux tampons de mémoire pour chaque Lettre, un chez l'abonné, et un autre dans le système frontal.

De plus, dans ce cas on peut prévoir des situations d'incompatibilité en ce qui concerne le facteur d'anticipation des Lettres sur un flot avec contrôle de flux. En effet on peut envisager qu'un abonné indique à son partenaire qu'il peut recevoir un certain nombre de Lettres car il dispose des tampons de mémoire suffisants, mais ceci n'implique pas que le système frontal en dispose d'autant.

### III.2.3 Division des Fonctions

Compte tenu des considérations précédentes, nous avons fait une décomposition des fonctions de la station de transport et nous les avons divisées en deux groupes. L'un, implanté sur l'ordinateur hôte constituant la Station Résiduelle, et l'autre, implanté sur le système frontal constituant la Station Externe.

### III.3 - LA STATION RESIDUELLE

La Station Résiduelle est une mince couche de logiciel implanté sur l'ordinateur hôte, dans notre cas, un IBM 360/67. Nous avons orienté nos efforts de façon que le logiciel exécuté par l'ordinateur hôte soit faible, et en particulier, sans aucune containte de temps réel.

#### III.3.1 Description des Fonctions

Le rôle principal de la Station Résiduelle est de fournir aux abonnés locaux les moyens pour accéder à la Station Externe. Les fonctions qu'elle exécute sont assez simples et peuvent se résumer par la liste suivante :

- Interface entre les abonnés locaux et la station Externe,
- Exécution des tâches de fragmentation/ré-assemblage des Lettres sur les tampons de mémoire fournis par les abonnés,
- Interface entre l'ordinateur hôte et la Station Résiduelle.

Dans la décomposition des fonctions de la Station de Transport, nous avons divisé les tâches de fragmentation ré-assemblage en deux parties : l'une s'exécutant sur la Station Résiduelle et l'autre s'exécutant sur la Station Externe.

La partie des tâches de fragmentation/ré-assemblage implantés sur la Station Résiduelle s'exécute en mode esclave, contrôlée par les commandes provenant de la partie implanté sur la Station Externe. Cette division permet de faire la fragmentation/ré-assemblage des Lettres sur les tampons fournis par les abonnés locaux tout en conservant le contrôle de ces

tâches sur la Station Externe.

### III.3.2 Implémentation de la Station Résiduelle

L'écriture du logiciel correspondant à la Station Résiduelle ne se trouve pas dans le cadre du projet PIAR. Cependant nous allons faire, à titre indicatif, une brève description du point de vue fonctionnel de deux possibilités d'implémentation.

#### III.3.2.1 Implémentation en Mode Usager

L'implémentation de la Station Résiduelle en mode usager utilise la même méthode que l'implémentation actuelle de la ST sous CP/67. C'est à dire, sur une machine virtuelle simulée par CP/67 et gérée par le système d'exploitation SYNCOP. Dans ce cas on peut reprendre la division du logiciel en trois processus : ECRITURE, LECTURE et ACCES; tout en éliminant les fonctions relatives au protocole de transport qui sont exécutées sur le système frontal.

ECRITURE consomme les blocs de données qui lui sont remis par les abonnés et les achemine vers le système frontal via le processus ACCES. Un bloc de données peut être une demande d'ouverture de flot, un télégramme ou un fragment de Lettre. Il exécute la tâche de fragmentation des Lettres qui ont une taille supérieure au maximum admis. Il n'envoie un fragment au système frontal que s'il reçoit une demande explicite de celui-ci.

LECTURE consomme les blocs de données provenant du système frontal via le processus ACCES. Il exécute la tâche de ré-assemblage des Lettres et ne les délivre aux abonnés correspondants que si le système frontal le lui indique. Un bloc de données peut être un télégramme, un fragment d'une Lettre ou une demande d'ouverture de flot.

ACCES fait l'interface entre le système frontal et la Station Résiduelle. Il consomme les blocs de données produits par le processus ECRITURE et les produit vers le système frontal. D'autre part il consomme les blocs de données reçus du système frontal et les produit vers le processus LECTURE.

L'interface avec les abonnés est fait de la même façon que dans l'implémentation actuelle de la ST décrite précédemment, tant pour les abonnés internes (abonnés-système) ou externes.

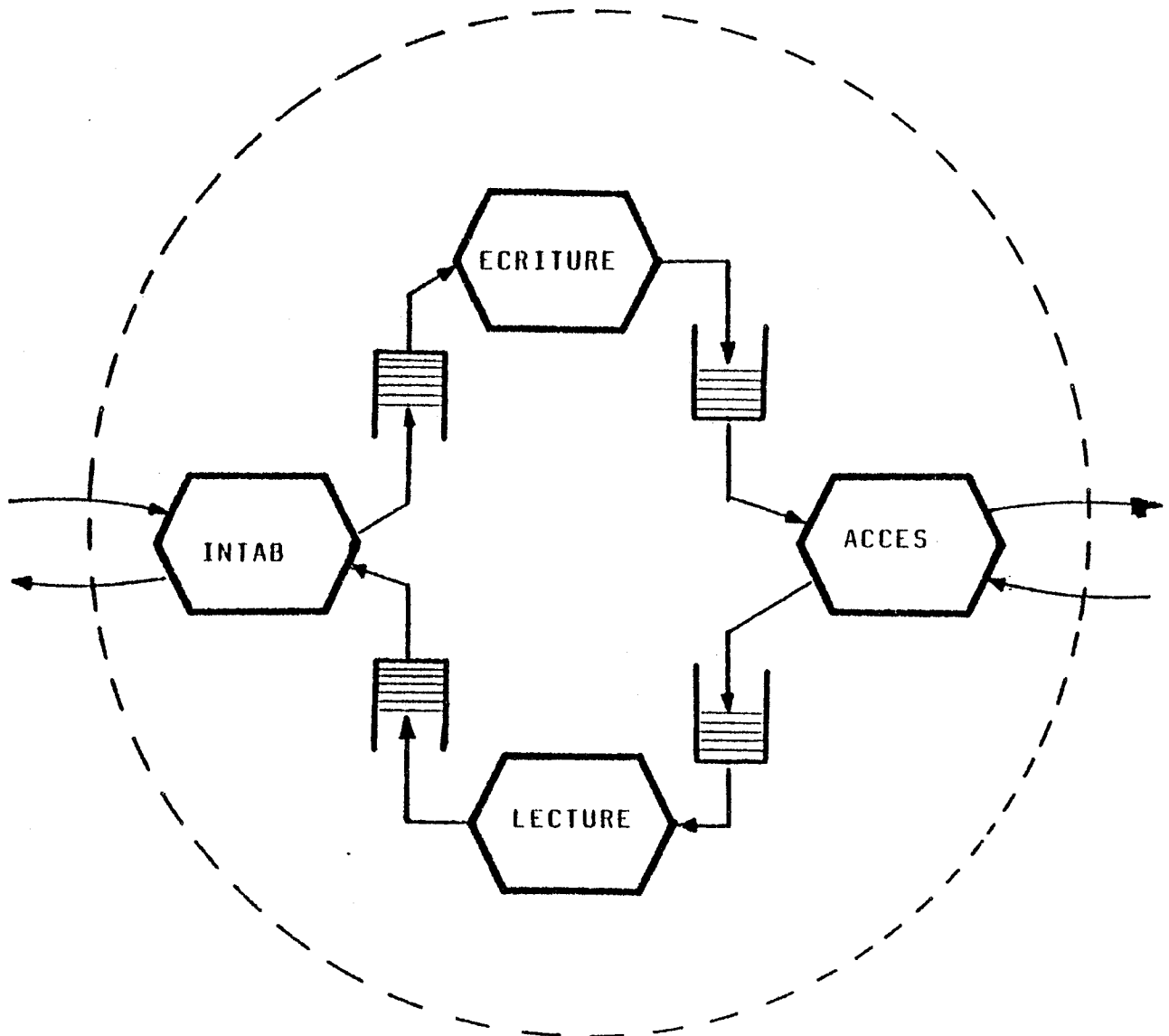


Fig. III-1 Implémentation de la Station Résiduelle - Mode Usager

### III.3.2.2 Implémentation Intégrée au Système

En observant la liste des fonctions exécutées par la Station Résiduelle, on note que les fonctions exécutées par la Station Résiduelle ressemblent à celles normalement associées à une méthode d'accès d'un périphérique quelconque.

En effet, compte tenu que toutes les fonctions propres au protocole de transport lui même sont exécutées sur le système frontal dans la Station Externe, nous pouvons dire que cette dernière est vraiment la station de transport et ce que nous avons défini dans un premier temps comme la Station Résiduelle n'est que la méthode d'accès à ce système frontal.

Vis à vis de l'ordinateur hôte, on peut faire l'analogie entre le réseau CYCLADES et une unité périphérique de disques. De ce point de vue, on peut considérer que l'abonné "voit" le système frontal, et à travers lui le réseau, comme étant une unité de disques qui lui est attachée.

Dans cette optique les équivalences suivantes s'appliquent :

- Un flot est considéré comme un fichier sur disque, et comme tel doit être ouvert par l'abonné avant d'y accéder, et doit être fermé à la fin.
- Une Lettre est considéré comme un enregistrement logique du fichier et est l'entité de données la plus petite adressable par l'abonné sur ce fichier.
- La fragmentation/ré-assemblage d'une Lettre est la conversion entre l'enregistrement logique et le, ou les, enregistrements physiques (secteurs) qui lui correspondent. Un fragment est alors considéré comme un secteur sur disque et est l'entité de données la plus petite adressable par la méthode d'accès.

- L'ouverture d'une porte (PI) afin d'y recevoir les demandes d'ouverture de flot distantes, peut être vue comme l'ouverture du fichier système qui contient le catalogue des fichiers existants.

Une différence est que, dans notre cas, l'unité de disques peut prendre l'initiative d'ouvrir ou fermer un fichier (flot) comme suite à une demande d'un abonné distant. Cependant on peut considérer que, au départ, le catalogue est vide et que les fichiers sont créés dynamiquement au fur et à mesure que les flots sont ouverts.

Une ouverture de flot par l'abonné local qui échoue se traduit par l'accès à un fichier inexistant. Une ouverture de flot par un abonné distant qui échoue est l'équivalent de vouloir créer un fichier sans avoir suffisamment de place sur disque.

Une ouverture de flot qui est refusé par la station de transport parceque elle n'a pas suffisamment de place (la limite du nombre de flots ouverts excédé) est l'équivalent d'un refus de création d'un fichier parceque le catalogue est plein.

A notre avis, il n'est pas possible d'utiliser le logiciel de la machine hôte sans le modifier. En particulier, il faut modifier ce logiciel afin d'éviter deux problèmes :

- il est impossible pour le système frontal, de reconnaître la nature des accès à l'unité de disques. En effet, les méthodes d'accès courantes ont été conçues pour gérer des unités de contrôle de disques "non intelligents". L'unité de contrôle reçoit seulement des commandes élémentaires (lecture, écriture, positionnement de tête de lecture, etc.). Elle ne connaît pas la correspondance entre ces opérations élémentaires et la comande de plus haute niveau qui les



origine (ouvrir un fichier, créer un fichier, consultation du catalogue, etc.).

- Il faut prévoir que, en réception, les blocs de données peuvent arriver en désordre. Or, ce problème n'est pas connu pour une unité de disques réelle.

### III.4 - LA STATION EXTERNE

Nous avons appelé Station Externe, la partie de la Station de Transport qui est implanté sur le système frontal. En fait, cette partie exécute toutes les fonctions relatives au protocole de transport et contrôle, d'une façon directe ou indirecte, toutes les fonctions de la Station de Transport.

#### III.4.1 Déscription Fonctionnelle

Vis à vis des abonnés, le fonctionnement de la Station de Transport est en général similaire à celui indiqué dans la notice de spécifications de réalisation de la Station de Transport portable que nous avons décrit dans le chapitre précédent (cf.II.3).

Les abonnés soumettent leurs requêtes BD vers la Station Résiduelle, qui les transmet à la Station Externe. Cette dernière les traite, génère les blocs de réponse BR correspondants et les transmet à la Station Résiduelle qui à son tour, les transfère vers les abonnés.

Afin que la Station Externe puisse contrôler les différentes fonctions sur chaque flot et chaque porte, elle contient les tables de contexte qui leur sont associées.

Nous décrirons par la suite la façon dont la Station Externe exécute et/ou contrôle les différentes fonctions de la Station de Transport.

##### III.4.1.1 Commandes de Négotiation

A la réception d'un bloc de demande BD d'un abonné local contenant une demande d'ouverture de flot, la Station Externe crée l'entrée dans la table de contexte correspondante et, si des services additionnels sont demandés, exécute le protocole de négociation de session. A la fin, elle génère le bloc de réponse

indiquant si le flot est ouvert ou non. Dans ce dernier cas, le bloc de réponse indique le motif pour lequel l'ouverture a échoué. La Station Externe détruit alors l'entrée dans la table de contexte qu'elle avait créée.

A la réception d'une demande d'ouverture de flot par un abonné distant, la Station Externe vérifie que l'abonné local ait une porte ouverte. Ensuite elle transmet un message à la Station Résiduelle qui à son tour informe l'abonné local. Ce dernier doit envoyer une demande d'acceptation d'ouverture vers la Station Résiduelle qui la transmet vers la Station Externe.

A la réception d'une commande de fermeture, la Station Externe ferme le flot, détruit la table de contexte associée et informe l'abonné local par le biais de la Station Résiduelle.

Les figures III-2 et III-3 montrent les séquences d'ouverture de flot à la demande d'un abonné local et à la demande d'un abonné distant.

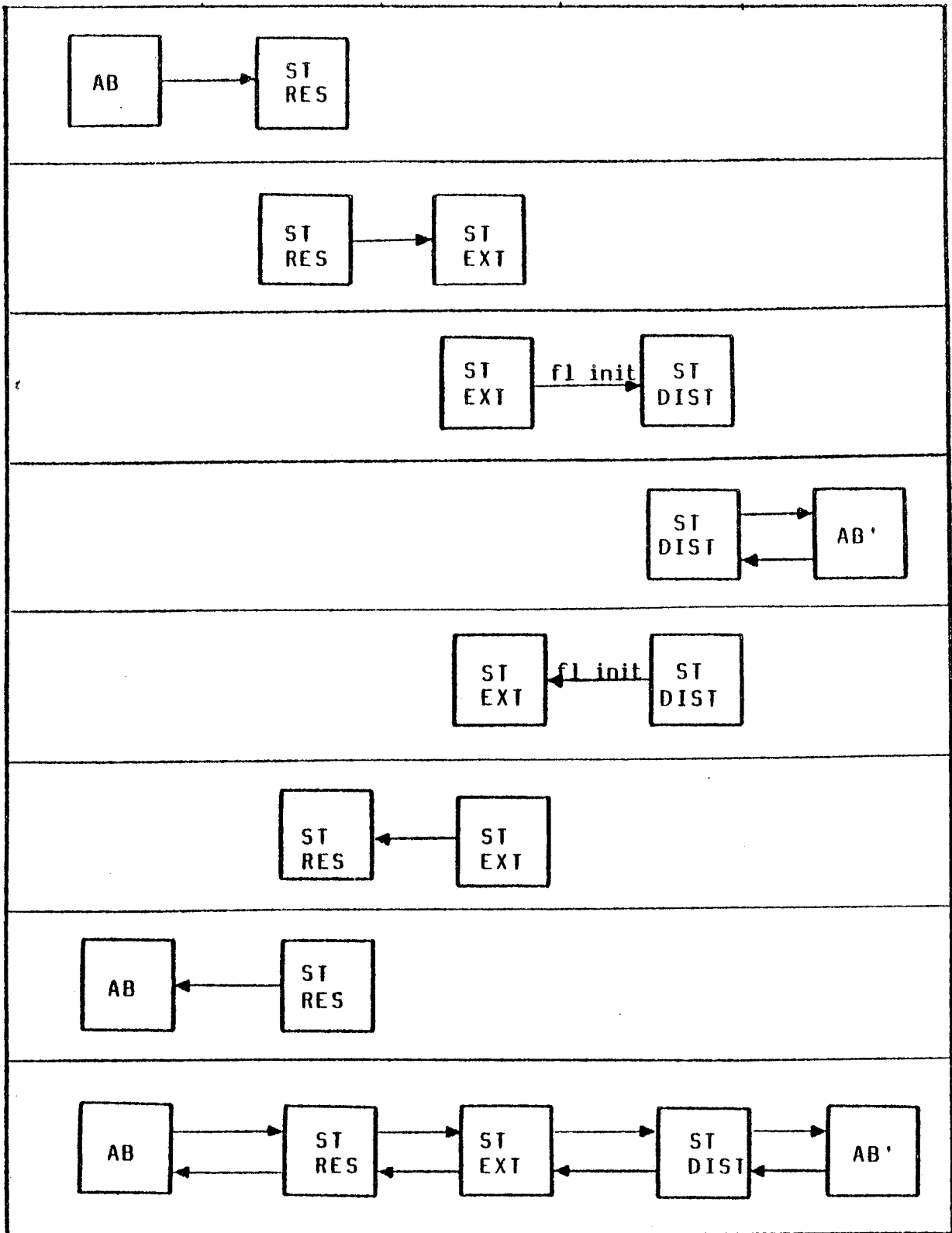


Fig. III-2 Ouverture de Flot par Demande Locale

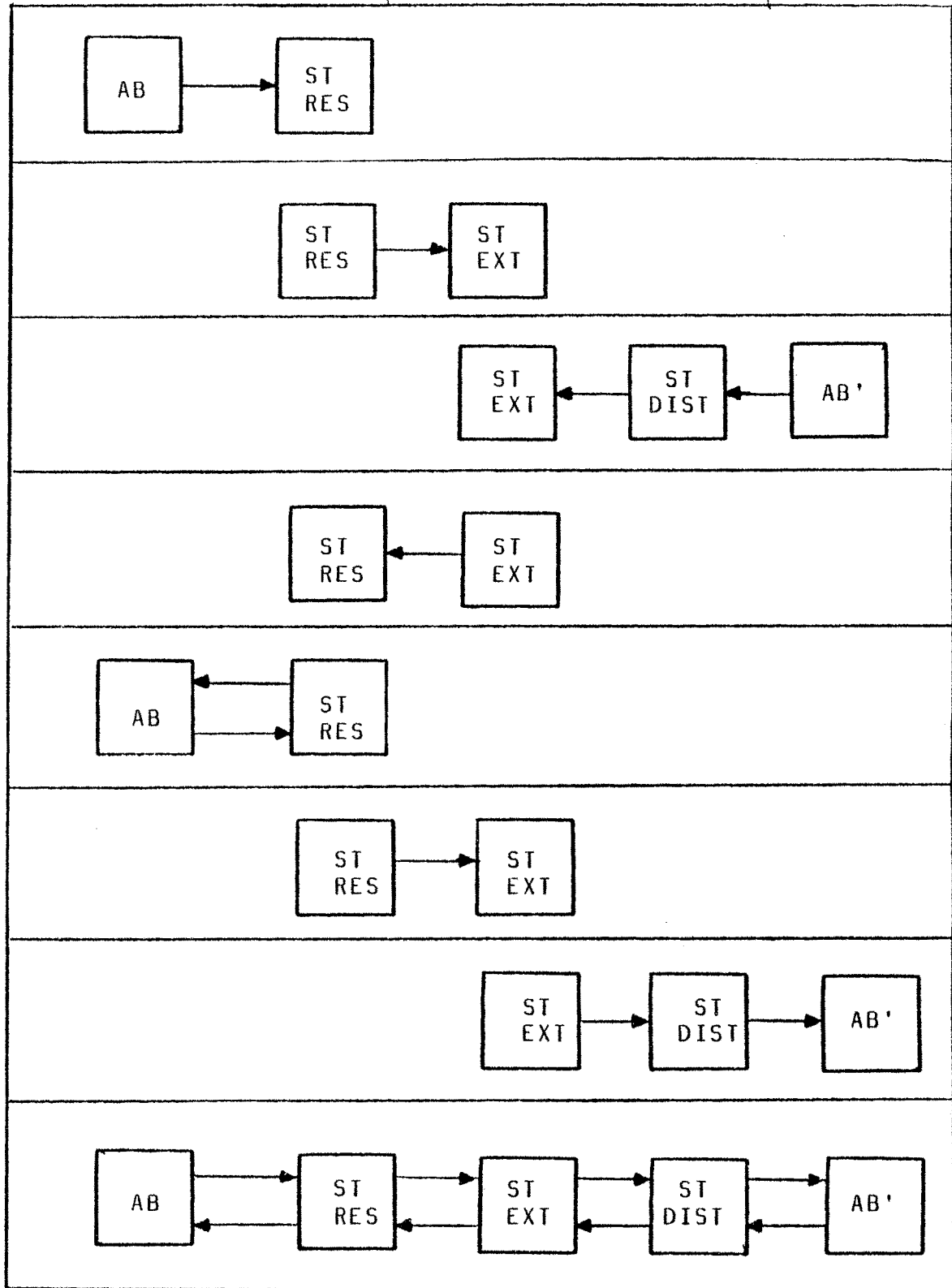


Fig. III-3 Ouverture de Flot par Demande Distant

### III.4.1.2 Fragmentation/ré-assemblage

Tel que nous l'avons mentionné précédemment, la fragmentation et le ré-assemblage des Lettres sont effectués par la Station Résiduelle sur les tampons de mémoire fournis par les abonnés. Cependant, le contrôle de ces tâches ainsi que les temporisations associées sont effectués par la Station Externe. C'est ainsi que ces tâches sur la Station Résiduelle fonctionnent en mode esclave, en se déroulant sous le contrôle des tâches de fragmentation/ré-assemblage qui s'exécutent sur la Station Externe.

Dans la Station Externe il existe une chaîne de bits (bit map) pour chaque Lettre en réception dans laquelle le bit "i" indique si le fragment "i" a été bien reçu. Dès que tous les fragments d'une Lettre ont été bien reçus, la Station Externe envoie un message à la Station Résiduelle indiquant qu'elle peut délivrer la Lettre à l'abonné destinataire.

La temporisation qui protège le ré-assemblage d'une Lettre est aussi exécuté dans la Station Externe. Si cette temporisation échoue, elle abandonne les fragments qui ont été reçus jusqu'à présent et attend que la SI distante envoie à nouveau la Lettre dès le premier fragment. Tout ceci reste transparent à la Station Résiduelle qui ne délivre une Lettre que lorsque la Station Externe le lui indique.

En émission, c'est la Station Externe qui décide quand un fragment d'une Lettre peut être envoyé. Elle envoie alors un message vers la Station Résiduelle en lui demandant d'envoyer un fragment spécifique d'une Lettre.

### III.4.1.3 Contrôle d'Erreurs

La Station Externe exécute les fonctions associées au contrôle d'erreurs sur un flot.

En réception, elle envoie à la ST distante, les acquittements correspondants aux Lettres bien reçues. Vis-à-vis de l'abonné local, elle assure le bon séquençement des Lettres car elle commande à la Station Résiduelle de délivrer une Lettre de référence "i" seulement si celle de référence "i-1" a déjà été délivrée.

En émission, la Station Externe enclenche une temporisation de réception d'acquiescement à la fin de l'émission de chaque Lettre. Si l'acquiescement n'a pas été reçu dans le délai prévu, la Station Externe commande à la Station Résiduelle de lui renvoyer la Lettre dès le premier fragment. Tout ceci reste transparent pour la Station Résiduelle qui ne libère pas le tampon de mémoire contenant la Lettre tant que la Station Externe ne le lui indique pas.

### III.4.1.4 Contrôle de Flux

La Station Externe fait la gestion des "crédits" des abonnés locaux et distants sur les flots sous contrôle de flux.

En émission, la Station Externe ne commande à la Station Résiduelle de lui envoyer les fragments d'une Lettre que si l'abonné local dispose de "crédits" chez son partenaire distant.

En réception, la Station Externe calcule si l'abonné local dispose des tampons de mémoire pour recevoir les fragments d'une Lettre, en fonction des "crédits" qu'il a manifesté disposer. Si ce n'est pas le cas, la Station Externe abandonne le fragment reçu et ne le transfère pas à la Station Résiduelle. D'autre part, la Station Externe se charge d'envoyer périodiquement vers

la ST distante les "crédits" disponibles chez l'abonné local.

#### III.4.1.5 Temporisations

La Station Externe lance, surveille et annule toutes les temporisations prévues dans les différents cas par le protocole de transport et le protocole de négociation. Dans chaque cas, elle exécute les actions prévues par le protocole. La Station Résiduelle ne se soucie donc plus d'aucune temporisation.

#### III.4.1.6 Télégrammes

Si un télégramme est reçu pour un abonné local, la Station Externe l'envoie vers la Station Résiduelle seulement si l'abonné a envoyé précédemment une demande explicite de réception des télégrammes. En cas contraire, la Station Externe abandonne le télégramme et ne le transmet pas à la Station Résiduelle.

#### III.4.2 Interface avec la Station Résiduelle

Le dialogue entre la Station Externe (SE) et la Station Résiduelle (SR) est fait à l'aide d'un ensemble de commandes bien défini. Ces commandes permettent, d'une part à la Station Externe de contrôler les différents processus de la Station Résiduelle et d'envoyer aux abonnés les blocs de réponse BR à leurs requêtes. D'autre part, ils permettent à la Station Résiduelle de transmettre à la Station Externe les requêtes des abonnés ainsi que les réponses aux commandes de la Station Externe.



La liste des commandes prévues pour cette interface est la suivante :

ENVSEBD - envoie vers la SE un BD d'un abonné,  
 ENVSRBR - envoie vers la SR un BR pour un abonné,  
 ENVSETG - envoie vers la SE un télégramme à transmettre,  
 ENVSRTG - envoie vers la SR un télégramme à délivrer,  
 ENVSRFR - envoie vers la SR un fragment de Lettre,  
 DEMSRFR - demande à la SR un fragment de Lettre,  
 ENVSEFR - envoie vers la SE un fragment de Lettre,  
 DELSRLT - commande à la SR de délivrer une Lettre,  
 LIBSRLT - commande à la SR de libérer une Lettre.

### III.4.3 Implémentation de la Station Externe

Tel que nous l'avons mentionné précédemment, la Station Externe est réalisée en dehors du système hôte, sur le système frontal PIAR. Cette implantation comporte tous les avantages mentionnés précédemment pour cette méthode (cf.I.2.3). Cependant, afin d'éviter les inconvénients associés (notamment le coût relativement élevé d'un mini-ordinateur ou d'un processeur spécialisé), nous avons décidé de concevoir un système frontal autour d'une architecture à plusieurs micro-processeurs en profitant de leur puissance de calcul et de leur faible coût.

C'est ainsi que nous avons conçu le système PIAR, (Périphérique Intelligent d'Accès au Réseau). Nous verrons par la suite l'implémentation de la Station Externe sur ce système.

Les figures III-4 et III-5 illustrent la distribution actuelle du logiciel d'accès au réseau et celle avec le système PIAR.

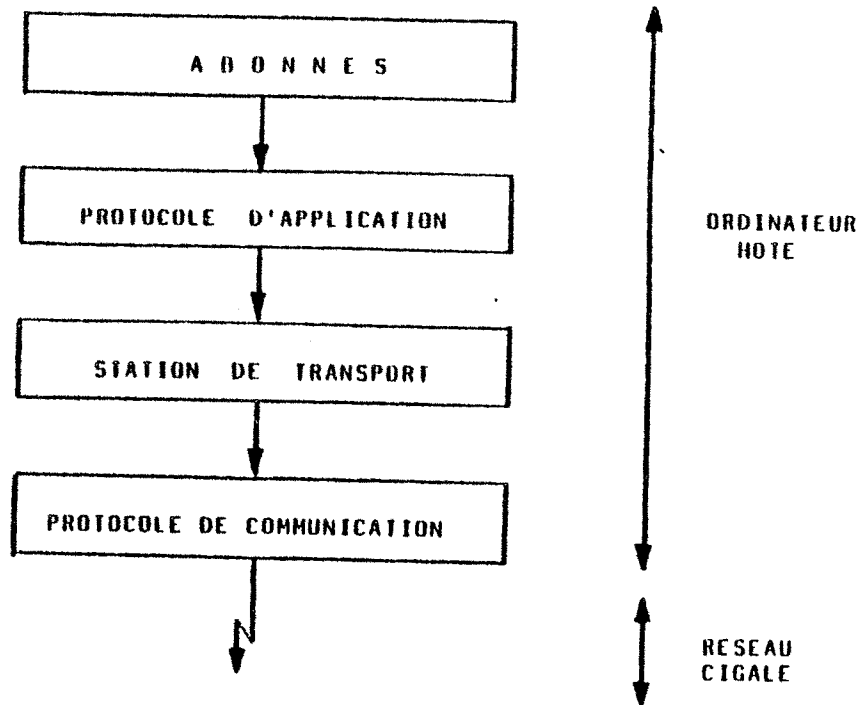


Fig. III-4 Distribution Actuelle du Logiciel d'Accès au Réseau

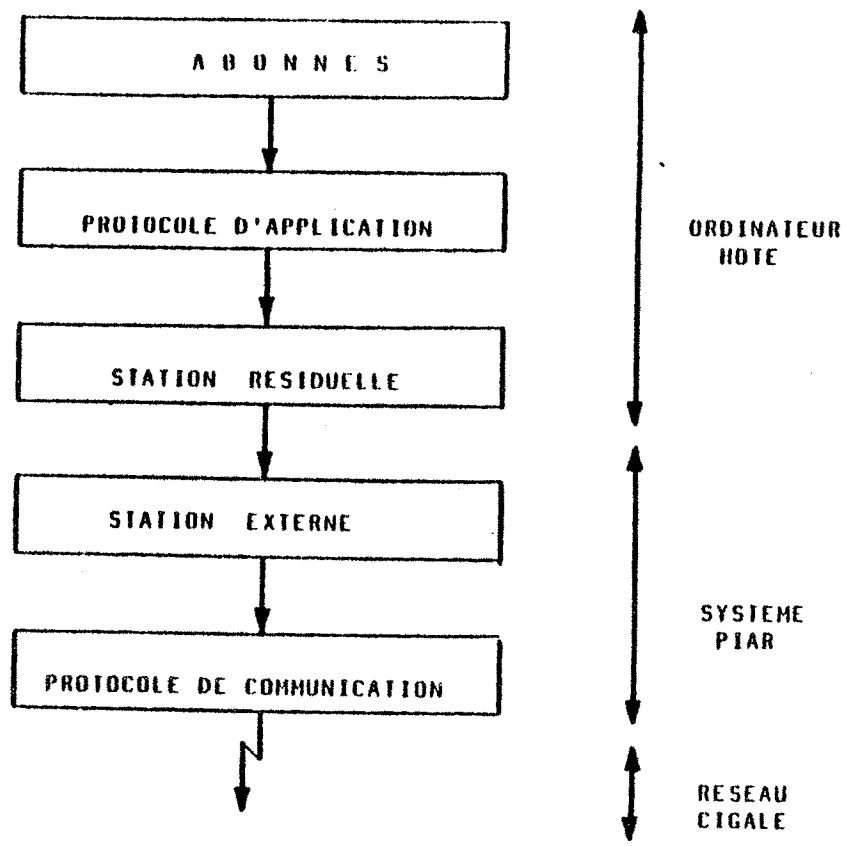


Fig. III-5 Distribution du Logiciel avec le Système PIAR

### III.5 - DEFINITION DU SYSTEME PIAR

Le but principal du système PIAR est de fournir des moyens matériels et logiciels simples et puissants permettant de connecter un ordinateur de la famille IBM 360/370 ou 303x au réseau CYCLADES.

L'idée principale qui a guidé la conception a été de décharger au maximum l'ordinateur hôte du logiciel réseau et de l'exécuter sur le système PIAR tout en respectant les normes d'interface avec le réseau.

Nous allons par la suite définir le système PIAR en fonction des différents points du cahier des charges que nous nous sommes fixés.

#### III.5.1 Description Fonctionnelle

Le système PIAR exécute trois ensembles de fonctions bien définis :

- les fonctions relatives à la Station de Transport,
- les fonctions relatives à la liaison avec le noeud du réseau de communication CIGALE,
- les fonctions relatives à la liaison avec l'ordinateur hôte lui même.

##### III.5.1.1 La Station de Transport

Le système PIAR exécute toutes les fonctions que nous avons définies pour la Station Externe (cf. III.4). C'est à dire qu'il exécute et contrôle, à l'aide du logiciel résiduel, toutes les fonctions associées à la Station de Transport.

Nous n'allons pas énumérer à nouveau ici la liste des fonctions ni leur description car elles ont été implantées telles que nous les avons définies précédemment.

#### III.5.1.2 Liaison avec le Réseau

La liaison du système PIAR avec le réseau est faite de manière "standard" au moyen d'une ligne de transmission de données synchrone. Cette ligne est gérée par la procédure de communication TMM-UCIG de façon à rester compatible avec les autres liaisons hôte-noued du réseau.

Le système PIAR fait la gestion de la ligne de transmission et exécute la procédure de communication ainsi que toutes les temporisations qui lui sont associées.

#### III.5.1.3 Liaison avec l'Ordinateur Hôte

Le système PIAR est connecté directement au bus du canal d'entrée/sortie de l'ordinateur hôte. Il se présente donc à cette machine, comme une unité périphérique capable de lecture et d'écriture avec son unité de contrôle.

Le système PIAR gère cette liaison selon la procédure et les conventions définies par le constructeur (24).

La communication entre l'ordinateur hôte et le système PIAR est donc faite en parallèle par octet. Cette liaison aurait pu être faite au moyen d'une ligne série de transmission de données en passant par une unité de contrôle du constructeur. Cependant, nous avons préféré que le système PIAR reste indépendant de la configuration du site et nous avons conçu une unité de contrôle banalisée pouvant gérer plusieurs types de périphériques (25).

De cette façon on décharge l'ordinateur hôte des contraintes de temps réel associées à la gestion d'une ligne série. La figure III-5 montre la liaison du système hôte au réseau selon la configuration actuelle et à travers le système PIAR.

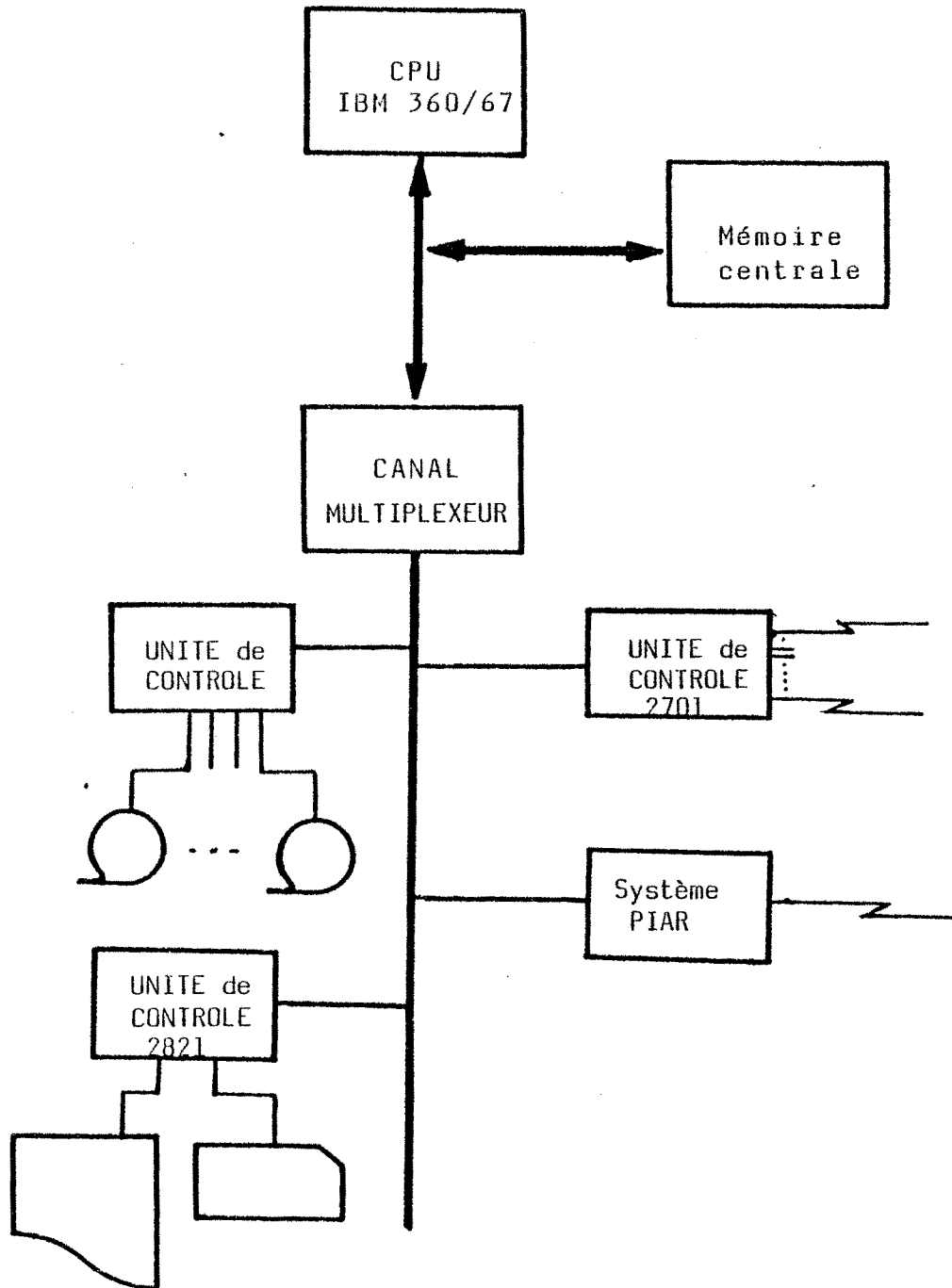


Fig. III-5 Connexion au Réseau CYCLADES

### III.5.2 Considérations de Compatibilité

Dans la définition du système PIAR nous avons tenu à conserver la compatibilité à tous les niveaux auxquels le système inter-réagit avec le milieu extérieur.

Cette compatibilité est respectée pour les spécifications électriques en ce qui concerne les interfaces matérielles et pour les spécifications fonctionnelles en ce qui concerne les interfaces logicielles décrites par des protocoles ou des algorithmes.

Cet ainsi que les spécifications électriques du bus canal IBM et de l'interface V24 (26) avec le MODEM de la ligne série sont strictement respectées. De même en ce qui concerne les restrictions temporelles des différents signaux dans chacune de ces interfaces.

Du point de vue fonctionnel, la décomposition de la Station de Transport en une Station Résiduelle et une Station Externe ainsi que l'implantation de cette dernière sur le système PIAR reste strictement transparente pour les abonnés locaux, les abonnés distants et les autres Stations de Transport du réseau.

### III.5.3 Considérations de Débit

Un des points importants que nous avons considéré dans la conception du système PIAR a été le débit de transfert et de traitement minimal à supporter afin de ne pas ralentir les communications hôte-réseau. En fait le débit des communications est déterminé par le débit de transmission sur la ligne série qui sert de liaison entre l'ordinateur et un noeud de CIGALE.

Actuellement, le débit maximal sur CIGALE est de 48 Kbits/sec. Nous avons prévu donc, que le système PIAR puisse travailler à un débit de 96 Kbits/sec de façon à tenir compte des améliorations ultérieures sur CIGALE.



CHAPITRE IVLES ARCHITECTURES A PLUSIEURS MICROPROCESSEURS

- 1 - Introduction
  
- 2 - Types d'Architecture
  - .1 Systèmes répartis
  - .2 Systèmes multi-processeurs
  - .3 Couplage entre processeurs
  
- 3 - Considérations de conception
  - .1 Choix de l'architecture
  - .2 La Structure d'interconnexion
  - .3 Caractéristiques des processeurs
  
- 4 - Le système PIAR
  - .1 Segmentation du logiciel
  - .2 Allocation des tâches
  - .3 Modules du système
  - .4 Communication entre modules
  - .5 Structure d'interconnexion
  - .6 Allocation des ressources





#### IV.1 - INTRODUCTION

L'idée d'utiliser plusieurs processeurs centraux afin d'améliorer les caractéristiques d'un système informatique n'est pas nouvelle. Les concepteurs ont toujours été motivés pour optimiser les performances globales d'un système en améliorant plusieurs paramètres, parmi lesquelles :

- Le débit de traitement,
- La souplesse,
- La disponibilité des différentes ressources,
- La fiabilité.

Malheureusement, le fait d'utiliser plusieurs processeurs centraux, n'implique pas une amélioration des performances dans un rapport comparable. En effet, une multitude de problèmes concernant l'interaction des processeurs, l'allocation des tâches, le contrôle et la synchronisation de processus parallèles pèse lourdement sur les performances globales du système. De même en ce qui concerne la fiabilité car, compte tenu de l'état de la technologie, la fiabilité des processeurs est très supérieure à celle de la mémoire commune.

Cette relation non-linéaire entre la performance et la fiabilité du système et le nombre de processeurs utilisés, ainsi que le coût élevé a découragé les concepteurs des systèmes et limité la réalisation de systèmes à plusieurs processeurs à quelques exceptions.

La technologie des circuits intégrés à grande échelle (LSI) a réduit le coût des processeurs centraux à des niveaux tellement bas, que l'utilisation des structures à plusieurs processeurs a pris une relance extraordinaire. Surtout en ce qui concerne la conception des ordinateurs de moyenne puissance. L'idée d'utiliser un microprocesseur par tâche ou par groupes de tâches

est très attrayante pour les concepteurs qui, compte tenu du prix de ces composants, ne craignent pas de sous-utiliser, voir gaspiller, la puissance d'un microprocesseur afin de décharger un autre processeur ou de simplifier le logiciel.

## IV.2 - TYPES D'ARCHITECTURES A PLUSIEURS MICROPROCESSEURS

Nous présentons par la suite une brève description des architectures à plusieurs microprocesseurs (37). Nous considérons ici les organisations où de multiples processeurs agissent sur de multiples ensembles de données, MIMD (36). Ces organisations, fournissent un certain parallélisme par l'exécution de tâches indépendantes sur des ensembles de données séparées. Pour obtenir des performances élevées dans cet environnement, il est nécessaire de distribuer les charges du système sous une forme équilibrée et de synchroniser proprement les processeurs dans leur traitement.

Deux types d'architecture peuvent se présenter :

- Systèmes répartis,
- Systèmes multi-processeurs

### VI.2.1 Systèmes Repartis

Les systèmes à architecture répartie, appelés aussi à multiples processeurs, se caractérisent par le fait que chacun des processeurs est dédié à l'exécution d'une fonction ou ensemble de fonctions d'un système unique distribué. Les processeurs peuvent être physiquement très proches les uns des autres ou distribués géographiquement.

Les tâches assignées aux processeurs et les actions qu'elles produisent doivent être connues à l'avance de façon à pouvoir répartir les fonctions du système parmi les éléments processeurs individuels. Ceci implique la segmentation du logiciel système en segments de programmes dédiés à chaque processeur. A chaque processeur sont attribués les variables et dispositifs qu'il doit contrôler.

Les avantages des systèmes repartis peuvent se résumer par la liste suivante :

- L'allocation statique des tâches minimise l'interaction entre programmes, et donc entre processeurs. Les communications entre processeurs sont généralement limitées à l'échange de messages ou de blocs de données par le biais des liens d'interconnexion entre les processeurs du système.
- L'allocation statique des ressources matérielles de chaque processeur simplifie la structure d'interconnexion entre les processeurs du système.
- Les processeurs peuvent être de types, modèles ou technologies différents. En général ils sont choisis d'après les caractéristiques qui s'adaptent le mieux aux tâches qu'ils doivent exécuter.
- La réalisation et la mise au point du matériel de chaque processeur peut être faite indépendamment des autres. Ceci s'applique aussi pour la plus grande partie du logiciel.

Les systèmes répartis présentent aussi certains désavantages, parmi lesquelles on peut mentionner :

- Il faut connaître à l'avance, et avec une certaine précision, la charge du système ainsi que les tâches qu'il doit exécuter. Dans le cas contraire, on risque d'utiliser les processeurs d'une façon inefficace à cause d'un mauvais choix dans le découpage du logiciel et du matériel.
- En raison de l'allocation statique des tâches et des ressources matérielles, ainsi que dans la

spécialisation des processeurs les systèmes repartis s'adaptent difficilement aux variations des charges ou des fonctions à exécuter qui n'étaient pas prévues dans la cahier des charges. En général, une modification dans les charges ou les fonctions du système risque d'entraîner une reconfiguration, voire une redéfinition du système.

- La défaillance ou la panne de l'un des processeurs spécialisés peut dégrader fortement les performances du système ou même le rendre inopérant. Ceci est dû également à l'allocation statique des tâches et ressources, car on ne peut pas ré-allouer dynamiquement les tâches exécutées par le processeur en panne sur un autre processeur. La pratique la plus courante est alors de dupliquer les processeurs spécialisés dans les tâches critiques.

#### IV.2.2 Systèmes Multi-microprocesseurs

Les systèmes multi-microprocesseurs utilisent un système d'exploitation unique qui, entre autres choses, alloue les tâches aux processeurs à fur et mesure qu'elles sont reçues. Cet allocateur peut être distribué, donc exécuté sur chacun des processeurs, ou centralisé sur un seul processeur (maître).

Les processeurs peuvent être banalisés et capables d'exécuter n'importe quelle tâche (cas symétrique), ou spécialisés dans l'exécution d'une tâche ou d'un groupe de tâches (cas asymétrique).

Les processeurs symétriques sont généralement utilisés dans les environnements où la charge du système change dynamiquement. Comme les processeurs sont identiques, une nouvelle tâche peut être allouée à n'importe quel processeur libre. De même, les tâches peuvent être ré-allouées si un processeur tombe en panne.

Les multi-microprocesseurs asymétriques sont composés de processeurs qui ont été configurés pour exécuter un ensemble particulier de tâches. Ceci risque de générer des files d'attente de tâches qui requièrent un même processeur spécialisé et entraîner aussi une dégradation des performances du système. toutefois, cela simplifie le système d'exploitation, qui devient alors un simple allocateur de tâches, sans se soucier de l'identification des tâches parallèles. Cette solution peut être améliorée en multipliant le nombre de processeurs spécialisés dans l'exécution des tâches les plus sollicitées.

Au fur et à mesure qu'un système multi-microprocesseur devient asymétrique, le nombre de tâches exécutées par des processeurs spécifiques augmente et certaines parties du système d'exploitation deviennent elles aussi spécifiques. Dans ces conditions, le système se rapproche d'un système réparti du type de ceux décrit précédemment.

Les avantages des systèmes à multi-microprocesseurs sont dérivés de l'allocation dynamique des tâches et des ressources. Ceci se traduit par des systèmes souples, modulaires et fiables qui ont un taux élevé de disponibilité vis-à-vis d'une nouvelle tâche ou demande de traitement.

Ces avantages se paient par un système d'exploitation et une structure d'interconnexion souvent complexes, et un volume élevé d'informations échangées entre les processeurs.

#### IV.2.2.3 Couplage entre Processeurs

En plus de la classification donnée précédemment, les architectures à plusieurs microprocesseurs peuvent être classées selon le type d'interaction entre les processeurs. Ainsi il existe un grand nombre d'architectures qui se trouvent entre les deux cas extrêmes : processeurs couplés de manière étroite et processeurs couplés de manière lâche.

On dit que deux processeurs sont couplés étroitement s'ils ont un haut degré d'interaction. Leurs algorithmes sont inter-dépendants et souvent l'un est en attente d'un événement ou d'une action causée par l'autre. Le cas extrême est celui où les processeurs partagent toute la mémoire qui contient les programmes. Les structures dont les processeurs sont couplés de manière étroite, sont généralement composées de processeurs bâtis autour des macro-composants LSI ou MSI. Ceci est caractéristique de systèmes à haut degré de parallélisme, tels que les architectures en "pipeline".

Dans les systèmes à couplage lâche, les interactions sont moins fréquentes et ne sont pas faites au niveau des programmes. En général, l'interaction se fait sous la forme de messages ou de blocs de données. Chaque processeur est un module, les relations entre modules étant du type producteur-consommateur. Il est fréquent dans ce type de structures que les processeurs partagent une mémoire en lecture/écriture accessible par tous les processeurs qui désirent dialoguer et qui sert donc au transfert des messages ou des données.



### IV.3 - CONSIDERATIONS DE CONCEPTION

Le progrès galopant des technologies à grande échelle d'intégration (LSI) permet la sortie sur le marché de circuits de plus en plus sophistiqués. Non seulement sous forme de processeurs centraux avec des architectures internes qui ressemblent à celles des mini-ordinateurs, mais aussi sous la forme de circuits périphériques chaque fois plus puissants et plus "intelligents". La puissance et la vitesse d'exécution de ces circuits augmente à chaque génération.

En plus, les fabricants ont fait des efforts visant la compatibilité et la simplicité d'interconnexion des circuits dans une même "famille", et souvent entre des "familles" de différents fabricants.

La réalisation d'un micro-ordinateur n'est plus une tâche inabordable aux non-initiés en électronique. En effet, les microprocesseurs ont déjà envahi le marché des amateurs, et il est relativement simple de définir une configuration particulière en interconnectant plusieurs circuits LSI comme les pièces d'un "méccano".

Cependant, dès que le concepteur s'attaque à la définition d'une architecture à plusieurs micro-processeurs, il doit résoudre un ensemble de problèmes inconnus dans les structures mono-processeur.

Parmi ces problèmes citons la définition de :

- la structure d'interconnexion,
- du contrôle des ressources du système,
- du mécanisme d'allocation des tâches,
- l'interaction entre processeurs,
- la procédure de récupération des pannes.

Pour chacun de ces problèmes il existe plusieurs solutions matérielles et/ou logicielles. L'approche prise par le concepteur pour résoudre chacun de ces problèmes aura un effet très profond sur un certain nombre des caractéristiques du système, parmi lesquelles :

- ses performances,
- sa sûreté de fonctionnement,
- sa souplesse,
- son coût.

#### IV.3.1 Choix de l'Architecture

Le premier choix que doit faire le concepteur concerne le type d'architecture du système. Son choix dépendra surtout des buts recherchés et des restrictions de l'environnement.

L'objectif peut consister à donner une ou plusieurs caractéristiques particulières au système, telles que :

- la banalisation des ressources et des processeurs afin de construire un système général,
- la spécialisation des processeurs afin d'exécuter des tâches précises de façon très performante,
- la modularité du système afin de pouvoir l'adapter à différentes charges et applications,
- la disponibilité du système afin qu'il puisse tolérer un certain nombre de pannes matérielles tout en minimisant les pannes fonctionnelles.

Le choix ne doit pas se limiter forcément aux deux types d'architectures mentionnées précédemment car il existe toute une variété d'architectures à plusieurs processeurs avec une ou plusieurs des caractéristiques souhaitées (37). Le rôle du concepteur est de définir une architecture dont les caractéristiques s'adaptent le mieux au cahier des charges.

#### IV.3.2 La Structure d'Interconnexion

La structure d'interconnexion des processeurs définit la topologie du système. Elle est étroitement liée aux options prises par le concepteur pour faire communiquer les processeurs entre eux, et aux restrictions de l'environnement.

Anderson et Jensen (28) ont proposé une classification arborescente des structures d'interconnexion à 4 niveaux. Ces niveaux sont définis en fonction de :

- la stratégie de transfert,
- la méthode de contrôle de transfert,
- la structure des voies de transfert,
- l'architecture du système.

Chacun de ces niveaux correspond à un niveau de décision du concepteur parmi les différents choix qui lui sont offerts. De ses choix sortira l'architecture du système ainsi que certaines caractéristiques des communications et du système en général.

Les deux premiers niveaux de décision concernent la politique générale d'interconnexion. Ces décisions tiennent compte des buts généraux du système ainsi que de la philosophie du concepteur. Elles définissent un certain nombre de caractéristiques du système comme :

- la modularité,
- la souplesse,
- la complexité logique,
- les risques de saturation,
- la reconfiguration sur panne.

Les troisième et quatrième niveaux de décision concernent les options d'implémentation des politiques définies précédemment. Elles tiennent compte des restrictions physiques et d'environnement du système. En particulier :

- la distribution géographique,
- le coût des voies de communication,
- le débit de transmission,
- le délai de réception des messages,
- la compatibilité avec les produits du commerce.

#### IV.3.3 Les Caractéristiques des Processeurs

Le choix du type, ou des types de processeurs, utilisés est lié, en principe, aux choix faits pour la structure d'interconnexion, les méthodes de communication inter-processeurs et les tâches à exécuter par le système.

Cependant, en pratique, les concepteurs font leur choix en fonction des facteurs relatifs à la commercialisation des microprocesseurs. En effet, d'après Lillen (29), les trois critères principaux qui interviennent dans le choix d'un microprocesseur sont, dans l'ordre d'importance :

- la disponibilité effective du produit,
- la réputation du fabricant,
- le support logiciel disponible.

Il existe aussi actuellement, une tendance à sous-utiliser un microprocesseur puissant, d'une nouvelle génération là où un "ancien" modèle, moins performant, aurait suffi (6). Cette sous-utilisation est liée à l'universalité des produits et est associée à la philosophie de que on peut tout faire, plus ou moins vite mais ça sera toujours assez vite pour une application quelconque si l'on dispose d'assez de puissance.

#### IV.4 - LE SYSTEME PIAR

Le système PIAR est bâti autour d'une architecture à plusieurs microprocesseurs. Cette architecture a été définie d'après les critères de décision que nous avons présenté dans la section précédente.

Nous allons par la suite décrire le système PIAR d'un point de vue architectural en tenant compte du contenu du cahier des charges que nous nous sommes fixés.

##### IV.4.1 Segmentation du Logiciel

Les fonctions exécutées par le système PIAR sont divisées en trois groupes bien définis :

- les fonctions liées au contrôle de la ligne série de liaison avec le noeud de CIGALE et à la procédure de communication qui la gère,
- les fonctions relatives au protocole de transport qui ont été définis pour la Station Externe,
- les fonctions liées à l'interface avec l'ordinateur hôte par l'intermédiaire de son processeur d'entrées/sorties (canal).

Chaque groupe de fonctions s'exécute de façon asynchrone et indépendante des autres. Les algorithmes des différentes tâches ne sont pas imbriqués. Une fois qu'une tâche est activée, son exécution ne dépend plus des autres. Nous pouvons donc considérer chaque groupe de tâches comme un module fonctionnel qui exécute un ensemble spécifique de fonctions.

Le système PIAR est donc une structure composée de trois modules fonctionnels qui coopèrent entre eux. Les mécanismes de dialogue entre ces modules sont de type producteur-consommateur entre tâches fixes.

#### IV.4.2 Allocation des Tâches

Dans la division fonctionnelle du système PIAR, deux des trois modules, concernent la gestion des liaisons du système avec le milieu extérieur. Notamment, la gestion de la ligne série vers le noeud de CIGALE et celle du bus du canal d'entrées/sorties de l'ordinateur hôte.

Ces liaisons sont matérialisées au moyen des interfaces spécifiques et bien définies qui requièrent des composants électroniques spécialisés afin de respecter les contraintes électriques et temporelles imposés par le milieu extérieur. Un module fonctionnel qui gère une liaison avec l'extérieur doit donc posséder en permanence les composants électroniques propres à l'interface qu'il gère.

Nous avons donc, dans le cadre du système PIAR, adopté une politique d'allocation statique des tâches aux processeurs. A chaque module fonctionnel, est alloué en permanence un microprocesseur spécifique qui exécute les tâches définies pour ce module.

#### IV.4.3 Modules du Système

L'allocation statique des modules fonctionnels aux processeurs ainsi que leur raccordement spécifique aux circuits spécialisés dont ils ont besoin matérialise la notion de module fonctionnel comme entité physique.

Dans notre cas, un module est un micro-ordinateur spécialisé dans l'exécution d'un ensemble de tâches. Il est composé d'un microprocesseur, de la mémoire morte contenant les programmes du module, de la mémoire vive pour les variables et les zones de travail, des portes logiques pour générer les signaux de contrôle du module et les circuits périphériques et les composants propres aux fonctions qui sont exécutées par ce module.

Le système PIAR est donc composé par trois modules :

- le module Contrôleur de Communication (MCC),
- le module Station de Transport (MST),
- le module Interface Canal (MIC).

#### IV.4.4 Communication entre Modules

Les relations entre les modules du système PIAR sont du type producteur consommateur. Un module produit un message ou un bloc de données qui est ensuite consommé par un autre module. Les modules sont donc couplés de manière lâche selon notre définition (cf. IV.2.3).

D'autre part, les modules ne sont pas distribués géographiquement car ils constituent une machine frontale physiquement contenue dans une armoire située parmi les autres organes périphériques du système hôte.

A partir de ces contraintes, nous avons décidé de faire communiquer les différents modules entre eux au moyen d'une mémoire commune unique à laquelle ont accès tous les modules. Cette mémoire est divisée logiquement en tampons, qui sont alloués dynamiquement aux modules.



La communication entre les modules est faite selon le principe des boîtes aux lettres. Chaque tampon de mémoire est considéré comme étant une boîte aux lettres à travers laquelle peuvent communiquer les différents modules.

#### IV.4.4 Structure d'Interconnexion

La structure d'interconnexion entre les modules du système est défini par les liens qui matérialisent la communication entre les modules. Le choix de ces liens dépend de la méthode choisie pour les faire communiquer.

Dans le cas du système PIAR, nous avons réalisé les voies de communication entre la mémoire commune et les différentes modules à l'aide d'un bus parallèle commun. L'allocation du bus est géré par un allocateur matériel spécialisé que nous allons décrire ultérieurement (cf.V.2.2).

Il est a noter qu'il n'existe aucun lien direct entre les modules. Toutes les communications sont faites de façon indirecte en passant par la mémoire commune. Nous avons ainsi évité d'avoir à réaliser un mécanisme d'interruption direct entre les modules (comme dans le système MICRAL (31), par exemple). De ce fait, la structure matérielle d'interconnexion est très simple.

Le système PIAR est donc un système à plusieurs microprocesseurs organisés selon une architecture répartie.

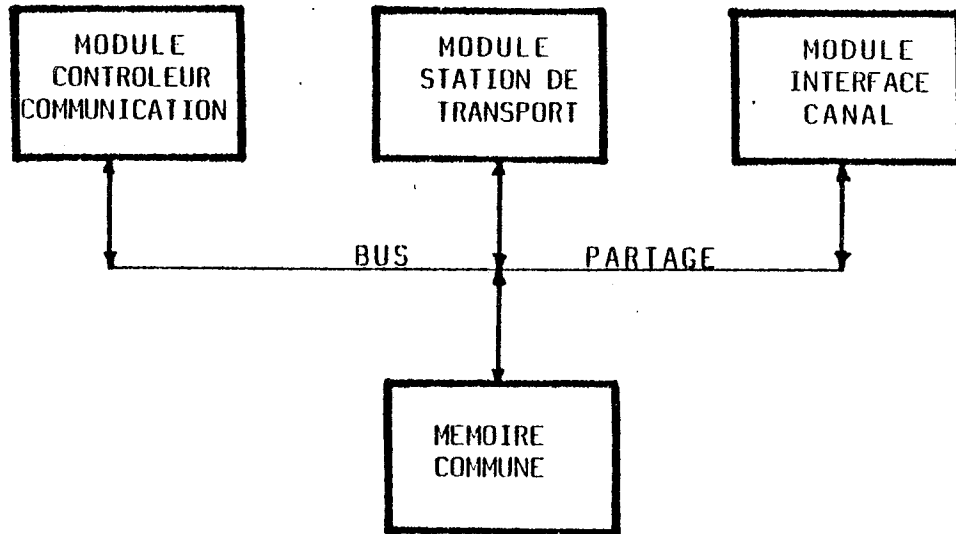


Fig. IV-1 Architecture du système PIAR

#### IV.4.6 Allocation des Ressources

Dans un système à microprocesseurs multiples existe un ensemble de ressources logicielles (programmes, données) et matérielles (bus, circuits périphériques mémoires) qui peuvent ou non être partagés.

Plus les ressources sont partageables, plus le système est souple et plus on peut exploiter ses possibilités au maximum. En contrepartie, chaque ressource partagée requiert un mécanisme de contrôle et de résolution de conflits. Un excès de ressources partageables entraîne des structures logicielles et matérielles complexes et une diminution globale des performances due à la surcharge introduite par l'autogestion.

Comme conséquence de l'allocation statique des tâches et de la constitution du système PIAR par des modules matériels, nous avons décidé de faire partager par les processeurs seulement les ressources nécessaires à la communication entre les modules.

Il existe alors deux niveaux d'environnements:

Un environnement interne à chaque module qui contient tous les circuits périphériques, une mémoire locale vive pour les données et les variables de travail, et une mémoire morte pour les programmes ainsi que les circuits spécifiques qui lui sont raccordés en permanence. Cet environnement, dit privé, est accessible seulement par les tâches exécutés par ce module.

Un environnement externe aux modules constitué par les objets partageables qui servent à faire communiquer les modules entre eux. Ce deuxième niveau dit commun, est composé de la mémoire et du bus parallèle qui sert de voie de liaison entre chaque module et la mémoire commune.

L'allocation de chacune des différentes ressources partagées peut être réalisé à l'aide de moyens matériels, logiciels ou une combinaison des deux. Le choix est fait de façon telle que le temps de réponse de l'allocateur à une demande soit une fraction du temps d'utilisation de la ressource par le demandeur. Chacun de ces objets partagés possède un allocateur qui le contrôle et gère les conflits dus aux requêtes simultanées des différents modules. Notons que ces allocateurs sont imbriqués car il faut que le demandeur ait alloué le bus et le tampon de mémoire afin de pouvoir y accéder.

Nous décrirons de façon détaillée les deux allocateurs du système PIAR dans le prochain chapitre (cf.V.2).

## CHAPITRE V

LE SYSTEME PIAR

- 1 - Hiérarchie du logiciel
  - .1 Propagation de l'activité
  - .2 Réalisation matérielle
  
- 2 - Ressources de communication
  - .1 Le Bus parallèle
    - .1 Allocateur de Bus
    - .2 Algorithme d'allocation
    - .3 Réalisation de l'allocateur de Bus
  - .2 La mémoire commune
    - .1 Allocateur de mémoire
    - .2 Algorithme d'allocation
    - .3 Implémentation de l'allocateur
    - .4 Mécanisme d'allocation
  
- 3 - Synchronisation entre modules
  - .1 Chemins de Synchronisation
  - .2 Le lien MIC-MST
  - .3 Le lien MST-MIC
  - .4 Le lien MCC-MST
  - .5 Le lien MST-MCC



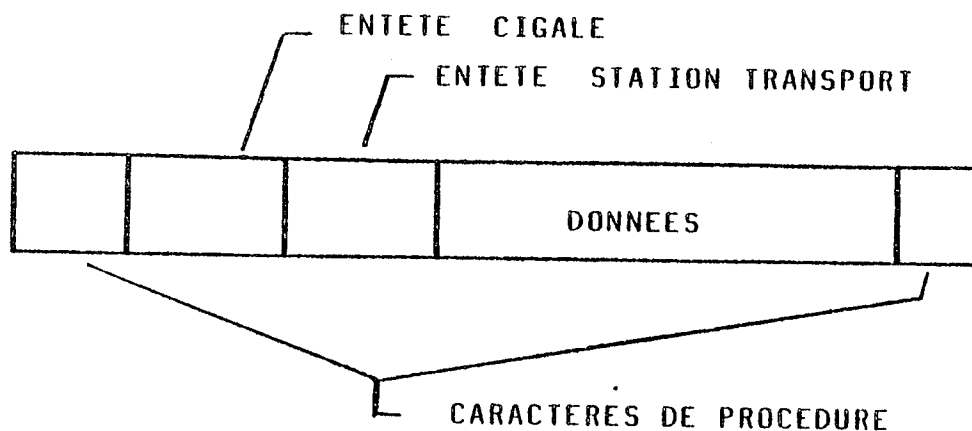
## V.1 - HIERARCHIE DU LOGICIEL

Dans le réseau CYCLADES, chaque message envoyé par les abonnés, est traité par tout un ensemble de protocoles. Ces protocoles sont empilés et chacun d'entre eux ignore l'existence des autres protocoles.

Un protocole est traité par une couche logicielle spécifique. Les différentes couches de logiciel constituent une structure hiérarchisée.

A chaque niveau de protocole est associé un ensemble de caractères et de messages de contrôle qui sont ajoutés en émission et éliminés en réception. Un protocole traite les messages et caractères de contrôle qui le concernent. Ce qui est considéré comme donné par un niveau de protocole peut être considéré comme contrôle pour un autre niveau supérieur de protocole.

L'empilement des protocoles génère un empilement de caractères de contrôle. Par exemple, la structure d'un fragment de Lettre émis par le contrôleur de communication est la suivante:



L'implémentation du logiciel de la station de transport sur le système PIAR fait apparaître deux nouveaux niveaux de protocoles :

- un gérant la connexion du système PIAR au canal de l'ordinateur hôte,
- un autre gérant la communication entre la Station Résiduelle et la Station Externe.

Ces protocoles s'intercalent dans la hiérarchie des autres protocoles du réseau. De même, ils restent transparents au reste du système car ils sont ignorés par les autres protocoles.

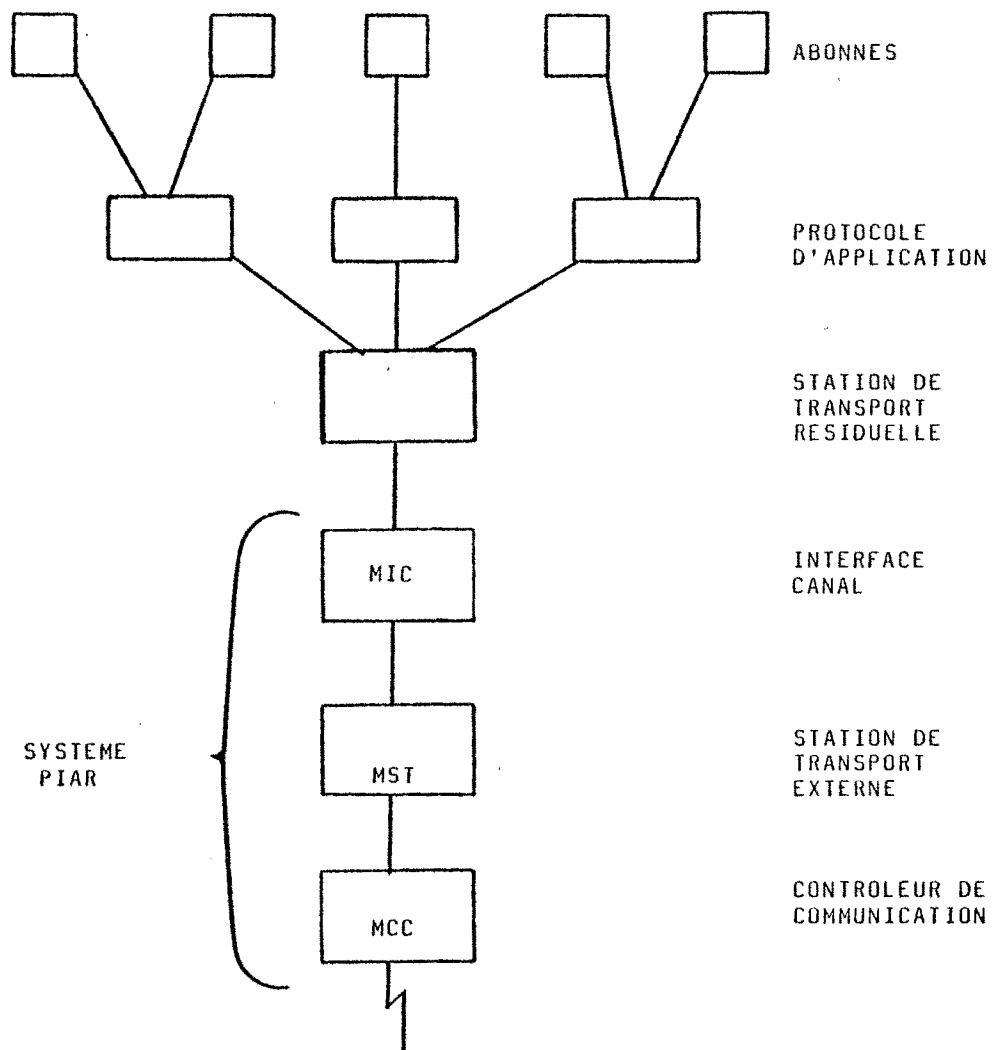


Fig. V-1 Hiérarchie des protocoles

### V.1.1 Propagation de l'Activité

Les relations entre les différentes couches de logiciel qui constituent la structure hiérarchisée dans le réseau CYCLADES, sont de type utilisateur-réssource (30).

L'activité d'une couche de logiciel dépend de facteurs internes et externes. Chacun de ces facteurs peut activer ou mettre en attente une couche quelconque de la structure.

Nous considérons qu'un facteur est interne à une couche si il dépend d'un événement propre à cette couche. Ainsi, une couche peut être activée parce qu'une temporisation qu'elle avait lancé précédemment a échoué. De même, une couche peut se mettre en attente d'une ressource qu'elle sollicite.

Un facteur est externe si il dépend d'un événement causé par les couches adjacentes (de niveau inférieur et supérieur) tel que les événements de synchronisation associés aux voies de communication entre les couches.

La représentation de la quantité d'activité dans la structure hiérarchisée peut être schématisé en forme d'une pyramide. Au sommet de cette pyramide se trouve le protocole de plus haut niveau (protocole d'application) et à la base est placé le protocole de plus bas niveau (protocoles du réseau de communication).

En émission, l'activité est propagée du haut vers le bas de la structure hiérarchique. Une action élémentaire à un niveau quelconque, induit une ou plusieurs actions élémentaires dans le niveau adjacent inférieur.



En réception, la propagation de l'activité est faite du bas vers le haut. Une ou plusieurs actions élémentaires d'un niveau quelconque induisent une action élémentaire dans le niveau adjacent supérieur.

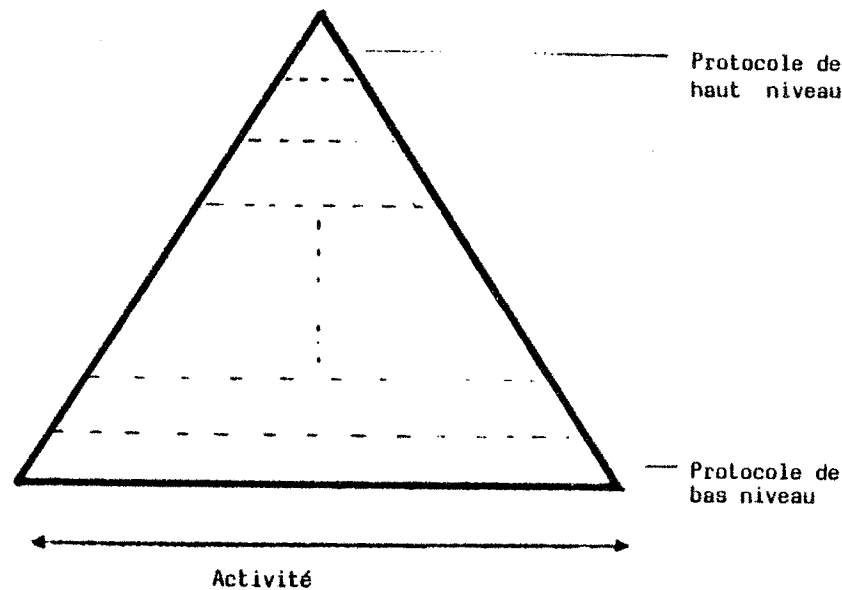


Fig. V-2 La Pyramide d'Activité des Protocoles

De même, dans le système PIAR, l'activité est propagée de couche en couche, donc de module en module. Il faut noter que dans le système PIAR, les transferts de données entre modules ne sont que de transferts logiques, car les tampons de mémoire sont affectés logiquement aux modules sans que les données qu'il contient soient déplacées physiquement. D'autre part, vis à vis des processus exécutés par les modules, l'allocation des tampons de mémoire aux modules constituent les événements d'activation de ces processus. L'exécution des différents processus du système PIAR est alors dirigé par les transferts des données.

En réception, le module Contrôleur de Communication (MCC) est activé par les messages provenant de la ligne série de liaison avec le noeud de CIGALE. Il traite les messages et caractères de contrôle qui lui sont pertinents et exécute les actions correspondantes qui sont prévues par la procédure de communication. D'autre part, il transfère vers le module Station

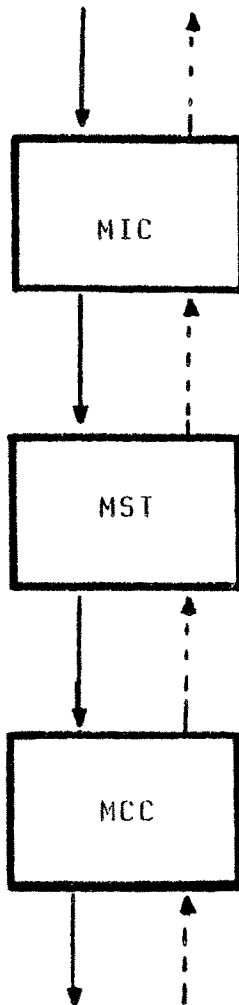
de Transport (MST) seulement les blocs de données qui ont été reçus sans erreurs.

Le module Station de Transport est activé par les blocs de données que lui transfère le module Contrôleur de Communication. A la réception d'un bloc, il vérifie la structure de son contenu et, s'il n'y a pas d'erreurs, il exécute les actions prévues par le protocole de transport. Le cas échéant, il transfère vers le module Interface Canal (MIC) le bloc de données correspondant.

Le module Interface Canal est activé à son tour par les blocs qui lui sont transmis par le module Station de Transport. Il déclenche ensuite un ensemble d'actions afin de prévenir l'ordinateur hôte qu'un bloc de données est prêt pour être envoyé vers la mémoire centrale. Une fois que l'ordinateur hôte lance l'opération de lecture sur le périphérique PIAR, le module Interface Canal envoie, à l'ordinateur hôte, le contenu du bloc de données, octet par octet, selon les conventions de la procédure d'échange avec le canal.

En émission, c'est le module Interface Canal qui est activé par les messages provenant de l'ordinateur hôte, qui à son tour, propage l'activité dans l'autre sens en envoyant les blocs de données vers le module Station de Transport.

Chaîne d'activité  
en émission



Chaîne d'activité  
en réception

Fig. V-3 Propagation de l'Activité

### V.1.2 Réalisation Matérielle

Il n'est pas toujours pertinent d'implémenter les structures hiérarchiques sous la forme d'une architecture matérielle qui lui ressemble. Dans le cas du système PIAR, nous avons réalisé à l'aide de seulement deux niveaux, la structure hiérarchisée du logiciel de trois niveaux.

Dans le premier niveau nous avons placé les modules fonctionnels. Chacun d'entre eux exécute le logiciel correspondant à l'un des niveaux de la structure logicielle. Au deuxième niveau nous avons réalisé le lien de communication qui sert à faire communiquer les modules entre eux. Ce deuxième

niveau est partagé par les modules du premier niveau et doit donc posséder un allocateur associé.

Les relations entre les modules ne sont donc pas du type utilisateur-réssource mais de liens fonctionnels entre les modules qui se trouvent à un même niveau (30).

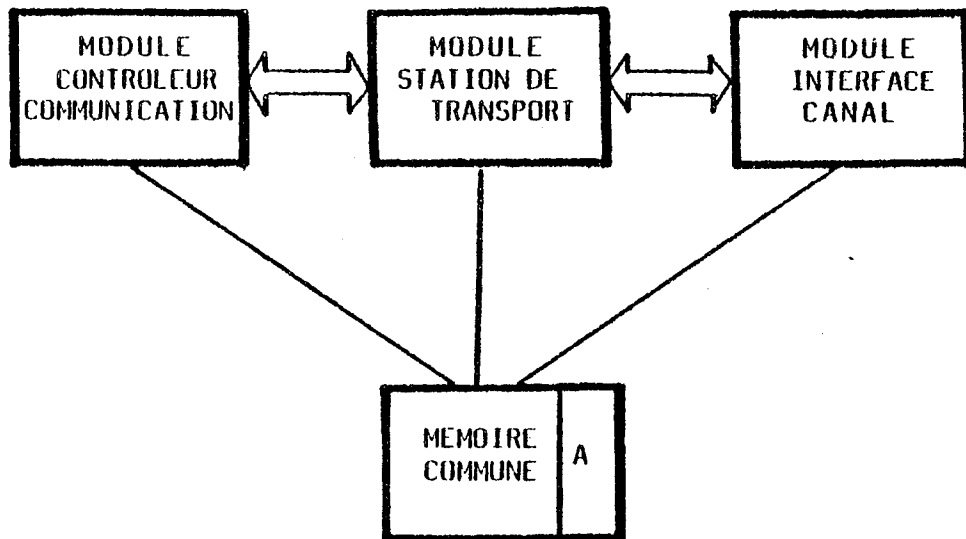


Fig. V-4 Organisation du Matériel

## V.2 - RESSOURCES DE COMMUNICATION

Le système PIAR possède deux ressources partageables par les différents modules : la mémoire commune qui sert à faire communiquer les modules entre eux et le bus parallèle, partagé dans le temps, qui fournit la voie d'accès à la mémoire commune.

Ces deux ressources, étant partagées, nécessitent des allocateurs afin de contrôler leur utilisation et ressoudre les conflits dus aux requêtes simultanées. Ces deux allocateurs ont été réalisés de façon telle que leur temps de réponse ( $T_{rep}$ ) à la requête d'un utilisateur reste une fraction du temps d'utilisation ( $T_{util}$ ) par le demandeur de la ressource en question.

$$T_{rep} \ll T_{util}$$

### V.2.1 Le Bus Parallèle

Le bus parallèle est le lien physique qui relie chaque module à la mémoire commune. Il est constitué par 21 lignes divisées en trois groupes fonctionnels distribués comme suit :

- 8 lignes bi-directionnelles de données,
- 12 lignes uni-directionnelles d'adresses,
- 1 ligne uni-directionnelle de lecture/écriture.

Ces lignes sont partagées par les trois modules. A un instant donné, un seul module contrôle l'état de chacune de ces lignes.

Du point de vue matériel, ces lignes sont du type "trois états". Au repos, si le bus n'est alloué à aucun des modules, chaque ligne se trouve dans le troisième état de haute impédance. Quand le bus est alloué à l'un des modules, c'est le module propriétaire qui impose l'état de chaque ligne, soit "1" ou "0"

logiques.

Chaque module peut donc relier son bus interne (données, adresses et ligne de lecture/écriture) au bus partagé. Cette liaison est matérialisée à l'aide de portes logiques du type "trois états". Ces portes assurent un isolement entre les deux bus tant qu'elles sont dans leur état inactif (haute impédance). L'état (actif ou inactif) de chacune de ces portes est contrôlé par l'allocateur de bus.

Il faut noter que seules les communications module-mémoire commune sont permises. Même si tous les modules sont connectés au même bus, il n'est pas possible de faire des échanges directement entre modules, (par exemple, des accès directs dans la mémoire locale d'un module à partir d'un autre module).

#### V.2.1.1 L'Allocateur de Bus

L'allocateur de bus régit l'allocation du bus partagé aux différents modules du système. Le bus est alloué à un module pendant le temps où il effectue un accès élémentaire à la mémoire commune (soit en lecture ou en écriture). Les accès multiples de type lecture, modification et écriture ne sont pas permis.

En général, le temps d'un accès élémentaire est un peu supérieur au temps d'accès de la mémoire commune (de l'ordre de 500 nano-secondes). Nous avons donc réalisé un allocateur de bus câblé avec un temps de réponse de quelques dizaines de nano-secondes afin de respecter l'inégalité mentionnée précédemment (cf. V.2).

#### V.2.1.2 Algorithme d'Allocation

L'allocateur de bus doit résoudre les conflits introduits dans le cas où deux modules demandent simultanément l'accès au bus partagé. Pour résoudre ces conflits, nous avons adopté une politique d'allocation selon une priorité statique. Chaque module

à une priorité fixe en fonction de son taux d'accès aux données contenues en mémoire commune. Ainsi nous avons donné la priorité maximale au module Interface Canal, ensuite au module Contrôleur de Communication et finalement au module Station de Transport.

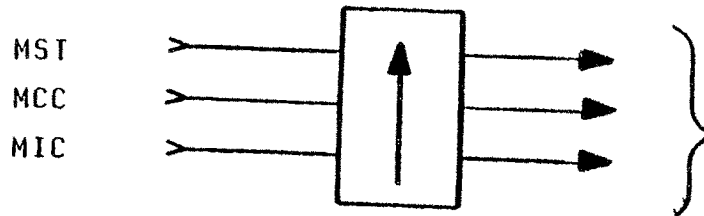


Fig. V-4 Priorité des Modules

L'allocateur de bus est donc un organe centralisé recevant les requêtes des utilisateurs.

La communication entre l'allocateur de bus et chaque module est faite à l'aide de deux variables booléennes. L'une, dénomée R, indique à l'allocateur si le module demande l'utilisation du bus. L'autre, dénomée A, indique au module si l'allocateur l'autorise à utiliser le bus. Bien entendu, ces variables ne sont pas partageables, mais sont variables propres à la communication entre un module et l'allocateur.

Il existe donc deux vecteurs booléens,  $R(1:N)$  et  $A(1:N)$ , où N représente le nombre de modules.

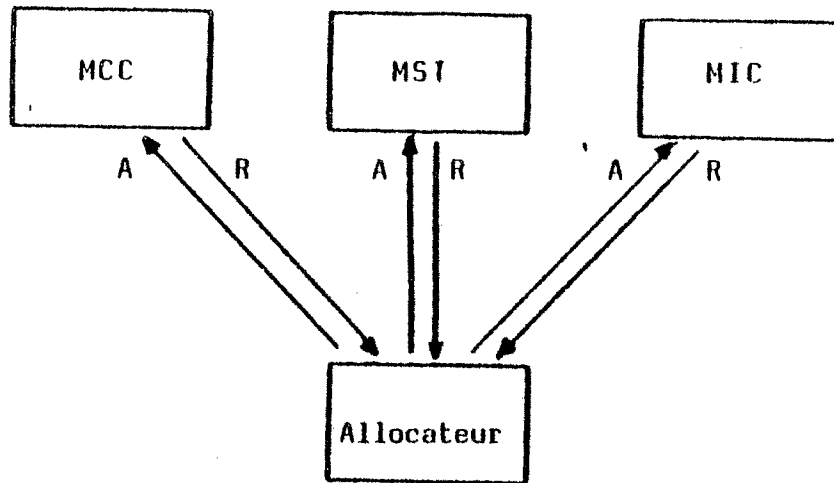


Fig. V-6 Communication des Modules avec l'allocateur de Bus

Les figures V-7 et V-8 montrent l'algorithme d'allocation et l'algorithme de demande et attente d'allocation des utilisateurs. Il faut noter que cet allocateur n'est pas équitable. Théoriquement il serait possible que la demande d'allocation d'un utilisateur de priorité faible ne soit jamais satisfaite si les utilisateurs de priorité supérieure font des requêtes en permanence.



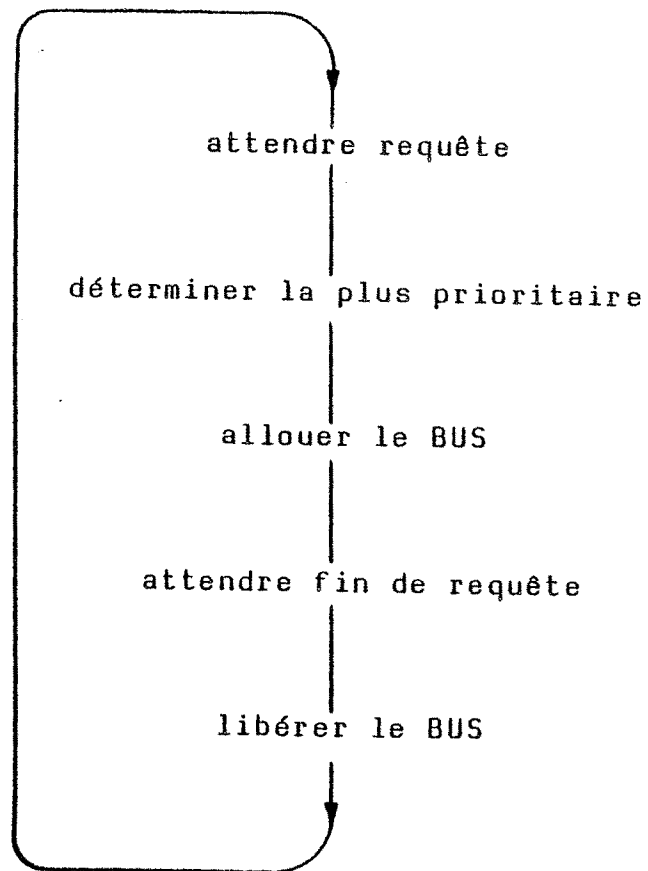


Fig. V-7 Algorithme d'allocation du Bus partagé

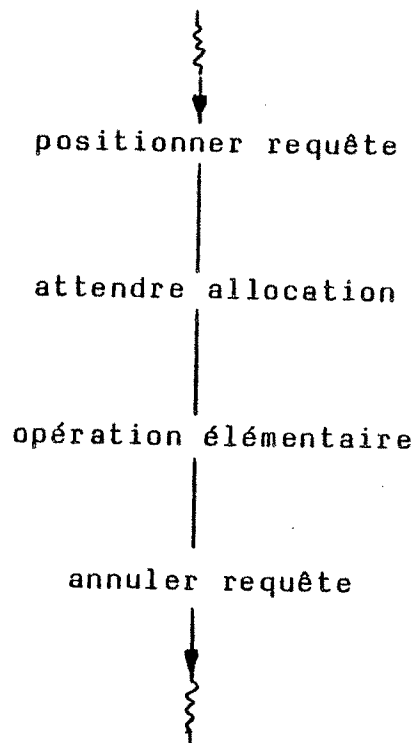


Fig. V-8 Algorithme des utilisateurs du Bus partagé

Cependant, en pratique, ce risque est inexistant car il faut tenir en compte la fréquence d'utilisation de la mémoire par les différents partenaires. D'après (30), le taux d'occupation de la mémoire par un processeur dépend :

- de la nature des informations qui y sont accédées :
  - . instructions et données,
  - . données,
  - . données globales,
- du temps de service de la mémoire en fraction du cycle de base du processeur.

La table suivante résume quelques valeurs approximatifs de ces variables.

Nature des informations	Temps de Service	
	cycle du processeur $1/n$	
instructions + données	100 %	$100/n$ %
données	25 %	$25/n$ %
données globales	3 %	$3/n$ %

Dans le cas du système PIAR, seul les données globales sont partagés dans la mémoire commune. Pour le microprocesseur Z80, la valeur de la variable "n" s'approche de 5. Le temps de service de la mémoire est alors inférieur à 2 %. On constate alors que, en pratique, il n'y a aucun risque de blocage d'un des modules de priorité faible par un autre de priorité plus élevé.

### V.2.1.3 Réalisation de l'allocateur de Bus

La réalisation pratique de l'allocateur de Bus nous a amené à modifier l'algorithme d'allocation de façon à tenir compte des contraintes temporelles et matérielles.

Comme les modules sont mutuellement asynchrones, et afin d'assurer l'indivisibilité des accès élémentaires à la mémoire commune, un organe de mémorisation des requêtes a été ajouté. Cet organe resynchronise les signaux de demande des utilisateurs avant de prendre la décision d'allocation (39).

D'autre part, les signaux de sortie de l'allocateur sont validées seulement après que les requêtes ont été mémorisés et que le circuit de priorité a eu le temps de stabiliser ses sorties.

Le fait d'introduire un organe de synchronisation des signaux asynchrones de requêtes des modules implique un risque de malfonctionnement des bascules de mémorisation. Ce phénomène, dit de "meta-stabilité d'une bascule", peut apparaître quand l'impulsion de chargement a une durée inférieure à celle nécessaire pour qu'elle prenne une décision sur la valeur à mémoriser (32). Il se manifeste par un délai anormalement long de la bascule pour stabiliser ses sorties.

L'entrée d'une bascule est essentiellement constitué par une porte "et" entre la signal asynchrone à échantillonner et l'horloge synchrone d'échantillonnage. Du fait de l'asynchronisme entre ces signaux, la sortie de cette porte peut être une impulsion de largeur arbitrairement petite (échantillonnage au moment d'une variation des entrées). La bascule peut prendre un état metastable si cette impulsion a une largeur comprise entre certaines valeurs dépendant du circuit. Cet état instable, dû au grand gain de la boucle de contre-réaction, peut rester pendant un temps indéterminé.

Plusieurs solutions ont été proposées pour éviter ce problème (32),(33). Nous avons adopté une solution dérivée d'une des solutions fondamentales proposées par Pechoucek (32). Cette solution diminue le risque de la meta-stabilité en contrôlant l'horloge d'échantillonnage par les signaux asynchrones.

Dans le cas de l'allocateur de bus, le problème réside dans le fait que les signaux de requête des modules sont mutuellement asynchrones. Nous avons donc choisit de générer une impulsion d'échantillonnage à chaque fois qu'il y a une transition des signaux d'entrée. De cette façon on assure que l'impulsion d'échantillonnage arrive après de la transition de la signal d'entrée, donc quand elle est stable (voir la figure V-10).

Dans ces conditions, une possibilité de que la bascule tombe dans l'état metastable est que il y ait une transition d'une autre signal d'entrée (requête) juste à la fin d'une impulsion d'échantillonnage. Cependant, cette nouvelle transition va à son tour générer une autre impulsion d'échantillonnage, ce qui force à la bascule de reprendre un état stable normal (voir la figure V-11).

D'autre part nous avons inhibé l'échantillonnage pendant que le bus est alloué afin d'assurer l'indivisibilité d'un accès élémentaire.

L'algorithme d'allocation devient alors :

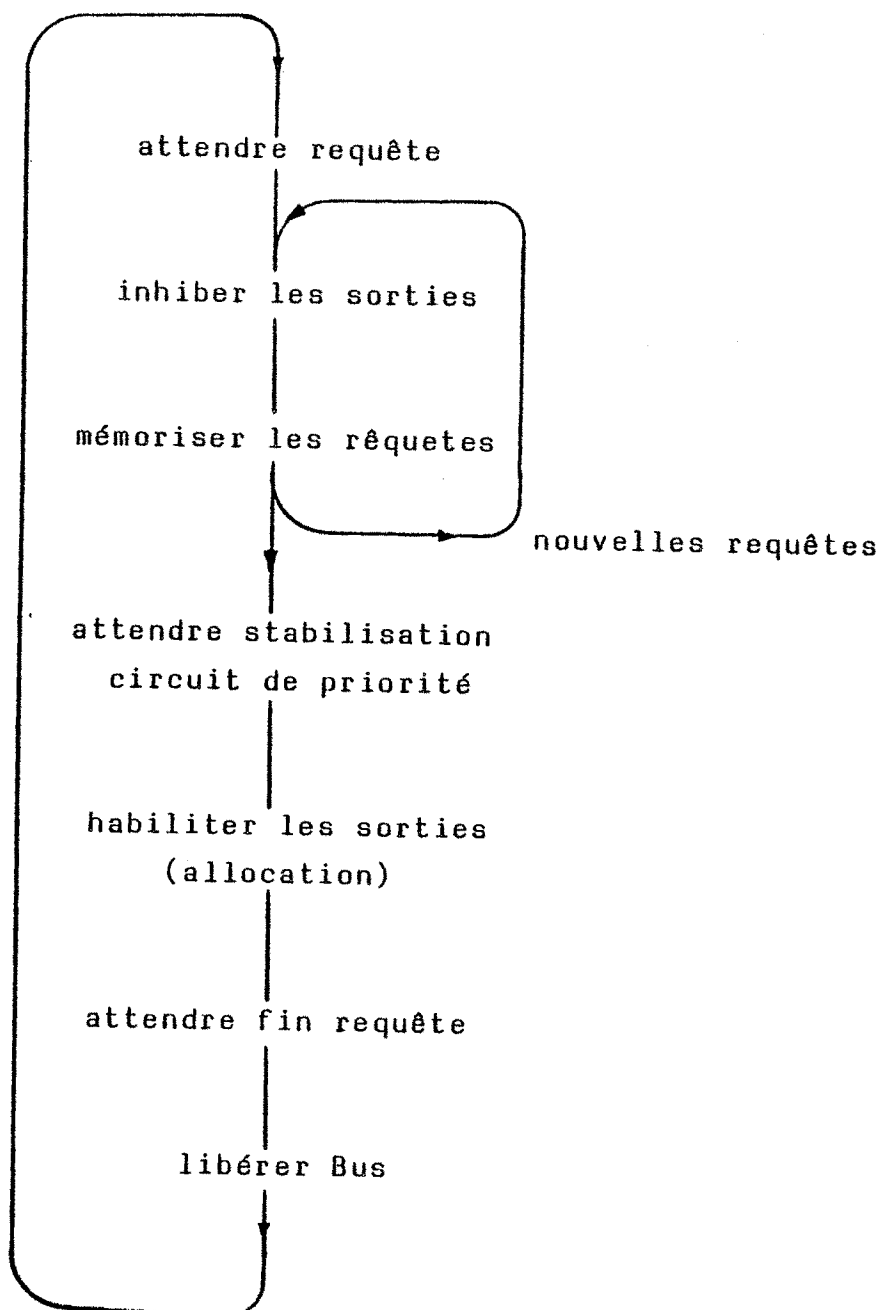


Fig. V-9 Algorithme de l'allocateur de bus réalisé

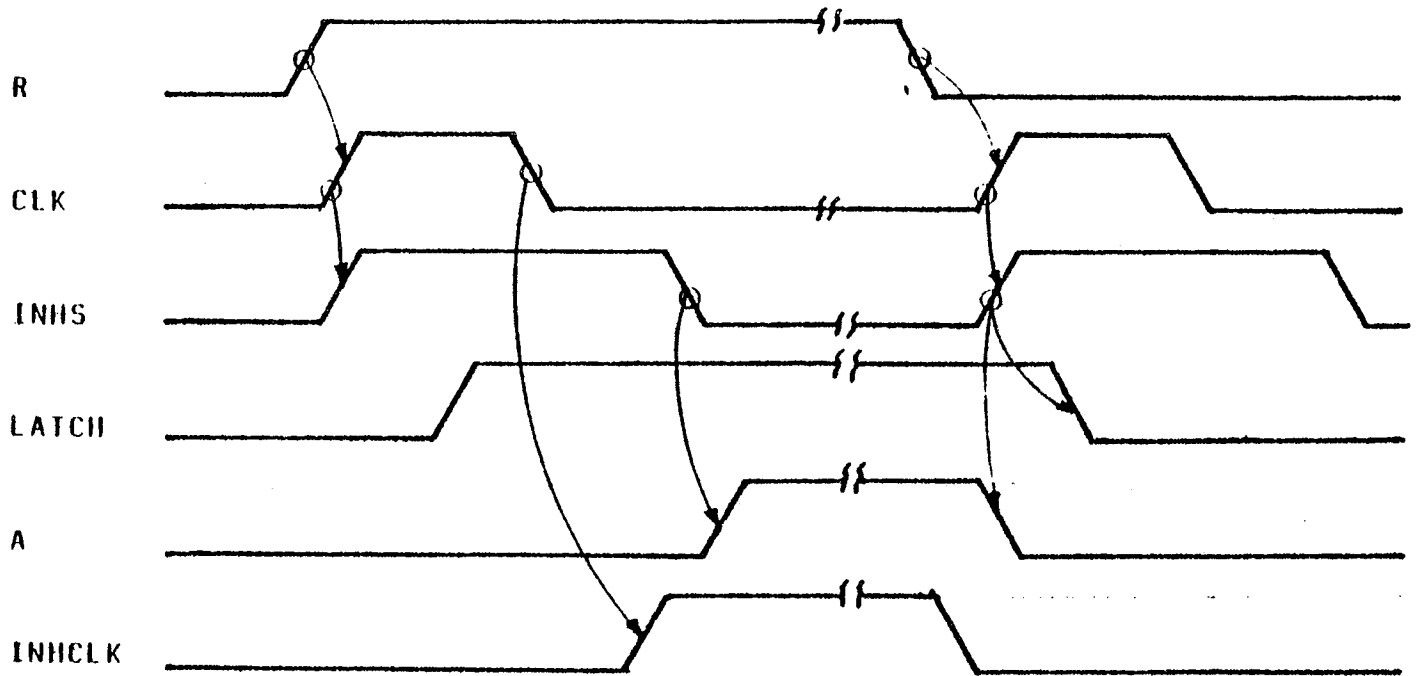


Fig. V-10 Diagramme des signaux d'une requête

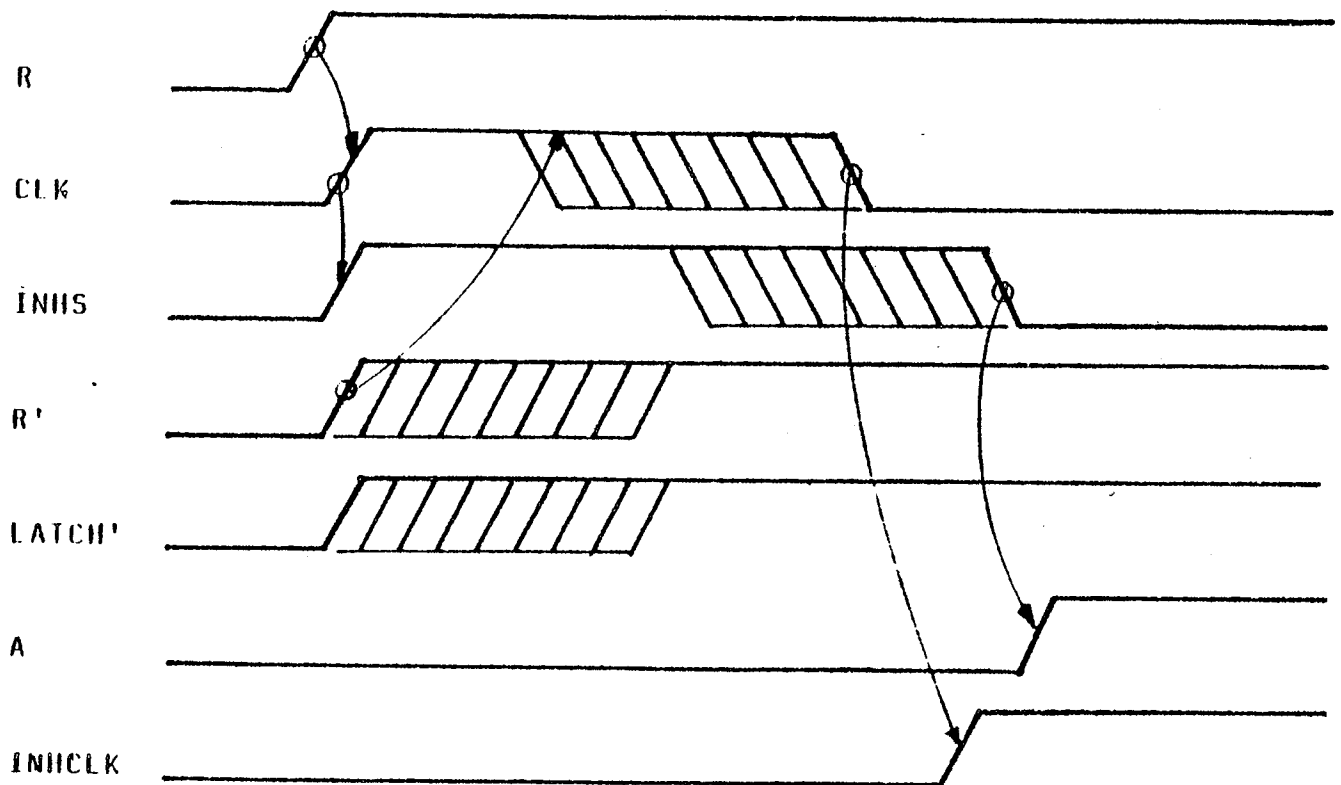


Fig. V-11 Diagramme des signaux des requêtes multiples

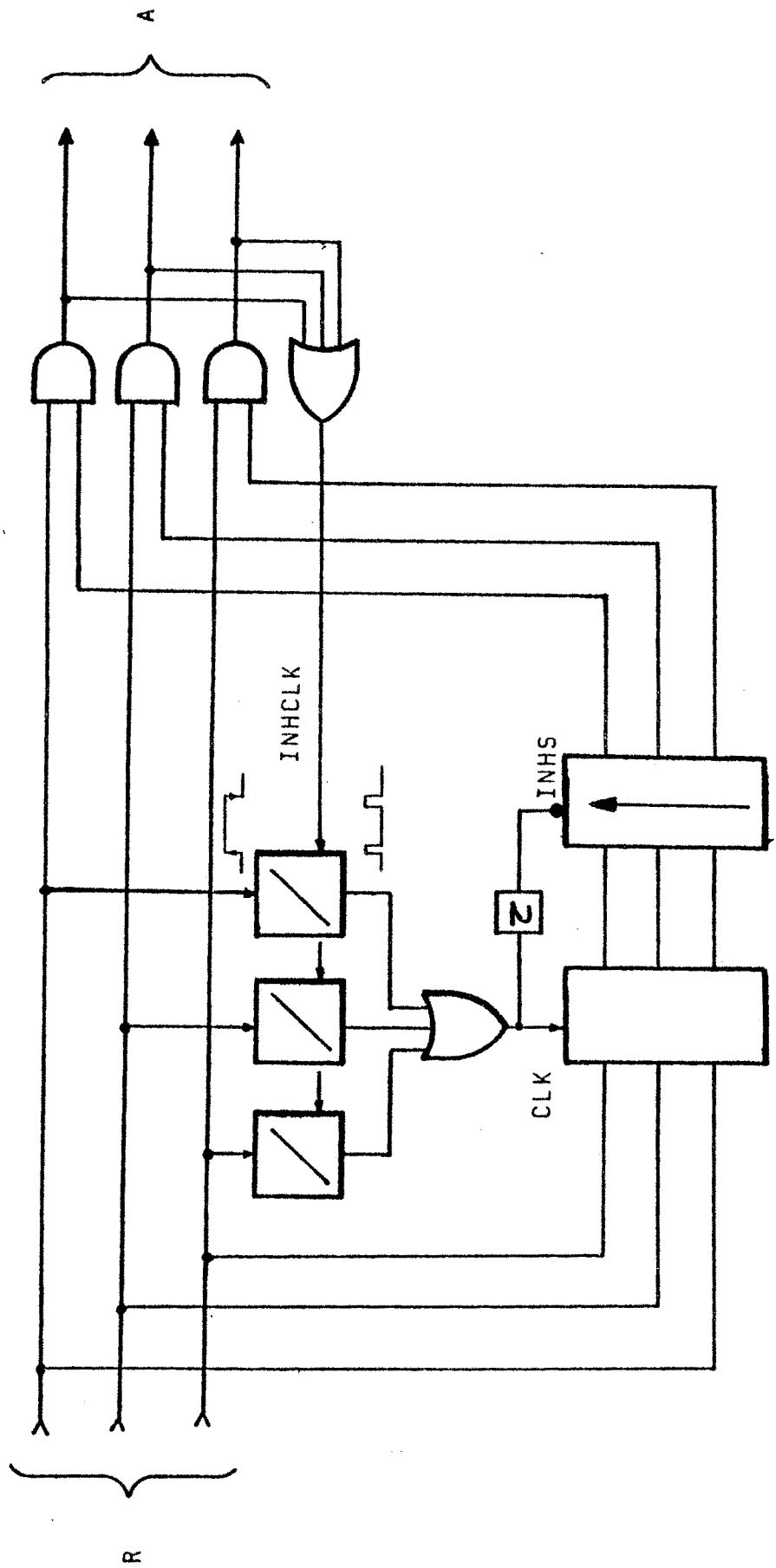


Fig. V-12 Schéma de l'allocateur de Bus

## V.2.2 La Mémoire Commune

Le mémoire commune est le seul lien utilisé par les modules du système pour communiquer entre eux. Elle est divisée logiquement en 10 tampons égaux de taille fixe. Chaque tampon peut contenir dans sa totalité un des messages échangés par les modules plus l'information de contrôle correspondante.

Ces tampons de mémoire sont utilisés par les modules comme des boîtes aux lettres. Les modules travaillent directement sur ces tampons qui matérialisent la notion de lien fonctionnel entre les modules.

### V.2.2.1 L'Allocateur de Mémoire

L'allocateur de mémoire regit l'allocation des tampons de mémoire aux différents modules du système.

Le temps d'utilisation d'un tampon en mémoire commune par un module est très variable et en fait dépend du contenu du tampon et des traitements à effectuer par le module propriétaire. Cependant, afin d'avoir une idée de l'ordre de grandeur du temps moyen d'utilisation, on peut appliquer les réflexions suivantes :

- D'une part, nous avons vu précédemment que le taux d'utilisation de la mémoire commune par un module, exprimé en pourcentage du temps du processeur, est très réduit (cf.V.2.1). Un module passe donc, la plupart de son temps à effectuer des opérations qui utilisent des ressources se trouvant dans son environnement privé (cf.IV.4.6). Si on considère que pour produire ou consommer un tampon, un module doit y faire plusieurs dizaines d'accès élémentaires, alors le temps d'utilisation d'un tampon par un processeur est supérieur à une milli-seconde.



- D'autre part, la chaîne d'activité du système (cf.V.1.1) nous indique que le débit global du système est limité par les échanges avec le milieu extérieur. Dans notre cas, la ligne série vers CIGALE et le canal de l'ordinateur hôte. De ces deux connexions, c'est en fait la première qui limite le débit global du système. Compte tenu du débit maximal prévu sur cette ligne, nous pouvons dire que le temps d'émission ou de réception d'un message est de plusieurs milli-secondes.

En fonction des réflexions précédentes, nous avons jugé qu'il n'était pas nécessaire de réaliser l'allocateur de mémoire dans le matériel. Il peut en fait être implémenté dans le logiciel tout en respectant l'inégalité mentionné précédemment (cf.V.2).

#### V.2.2.2 Algorithme d'Allocation

Il existe un grand nombre d'algorithmes d'allocation généraux pour la gestion d'une zone de mémoire partagée par plusieurs utilisateurs. Ces algorithmes sont en général assez complexes justement parceque ils sont généraux. D'autre part, dans la plupart des cas ils son centralisés et réquièrent l'existence d'un processus maître (allocateur) qui alloue les zones de mémoire aux utilisateurs.

Nous avons décidé de réaliser un allocateur de mémoire reunissant les caracteristiques suivantes :

##### Distribué

Les modules du système PIAR se trouvent a un même niveau dans la hiérarchie. Il n'est donc pas indiqué d'implanter l'allocateur de mémoire sur un d'entre eux car d'une part, il deviendrait maître par rapport aux autres modules et d'autre part, il ne se justifie pas de ajouter

un module spécialisé ou un microprocesseur au système pour qu'il exécute seulement la fonction d'allocateur de mémoire. Nous avons donc décidé de réaliser un allocateur distribué. Chaque module exécute l'algorithme d'allocation de façon indépendante et asynchrone des autres.

### Simple

Nous avons cherché à ce que l'algorithme d'allocation soit simple de façon à minimiser le volume de code nécessaire à son implémentation. L'écriture et la mise au point des programmes deviennent ainsi très simples. Dans cette optique nous avons réduit au minimum les variables partagées, donc critiques. D'autre part, nous avons évité d'utiliser des opérations indivisibles de type "test et modification", (test&set) car elles ne sont pas couramment disponibles dans les microprocesseurs que nous avons utilisés.

### Spécialisé

Il n'était pas notre but de faire des études approfondies afin de concevoir un algorithme d'allocation général. Au contraire, nous avons profité au maximum des conditions et cas particuliers à notre application afin de simplifier l'algorithme et son implémentation.

### V.2.2.3 Implémentation de l'allocateur

Dans le cas du système PIAR, les tampons en mémoire commune ne sont pas banalisés. Au lieu d'avoir un ensemble de tampons "libres" alloués aux modules au fur et mesure qu'ils le demandent, nous avons décidé de dédier un sous-ensemble de tampons à chaque lien fonctionnel entre modules. Il existe donc, 4 groupes de tampons :

- groupe A, dédié à la liaison MIC vers MST,
- groupe B, dédié à la liaison MST vers MIC,
- groupe C, dédié à la liaison MCC vers MST,
- groupe D, dédié à la liaison MST vers MCC.

Nous avons défini l'algorithme d'allocation en fonction de la chaîne d'activité du système PIAR qui détermine la façon dont y transite l'information. Chaque module consomme et produit dans un ensemble fixe de tampons. Les modules peuvent se donner des tampons mais ne peuvent s'en attribuer. D'autre part, un tampon de mémoire n'est jamais "libre". Il est toujours alloué à un des modules.

A l'initialisation du système, chaque producteur se voit alloué les tampons qui correspondent aux liens fonctionnels pré-établis. Un fois qu'un module à fini de produire dans un tampon qui lui appartient, il transfert (alloue) ce tampon au module consommateur. Le module consommateur reçoit le tampon et, une fois qu'il à fini les traitements correspondants, le transfert vers un autre module.

Un tampon de mémoire peut donc avoir trois états différents selon le module auquel il est alloué à un instant donné. L'évolution de l'allocation d'un tampon de mémoire peut être décrit à l'aide d'un automate d'états finis. Un état indique à qui est alloué un tampon à un instant donné. Les transitions correspondent aux allocations (transferts) faites par le module propriétaire à un instant donné vers un autre module.

Comme chaque groupe de tampons est dédié à un lien fonctionnel spécifique, il existe un automate différent décrivant l'algorithme d'allocation de chacun d'entre eux. Les figures V-11 à V-14 montrent les automates des 4 groupes de tampons.

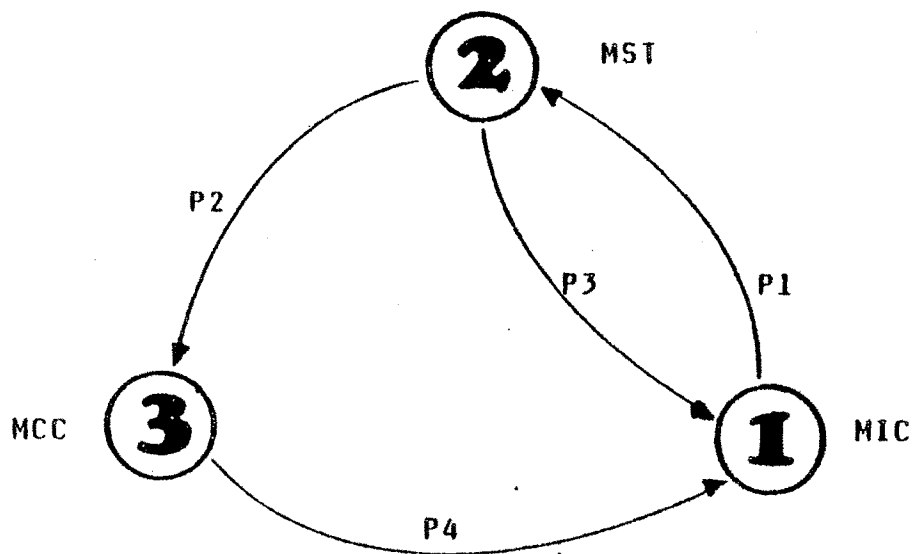


Fig. V-12 Automate des tampons du groupe A

Cet automate décrit la réception par le module MIC de paquets de données ou de contrôle provenant de la Station Résiduelle, leur analyse par le module MST et leur envoi au module MCC avec retour des tampons vides au module MIC.

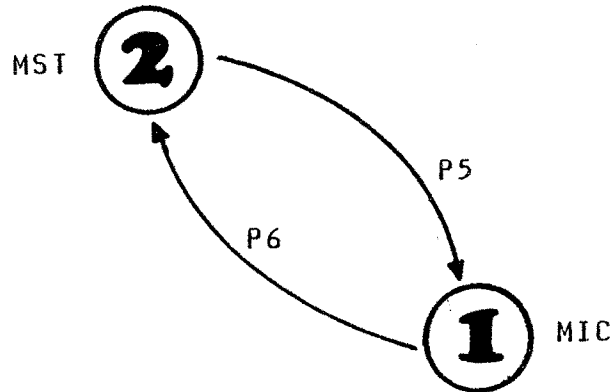


Fig. V-13 Automate des tampons du groupe B

Cet automate décrit le transfert des messages du module MST vers le module MIC et le retour des tampons vides au module MST. Sur ce lien sont envoyés les messages de contrôle de la Station Externe vers la Station Résiduelle.

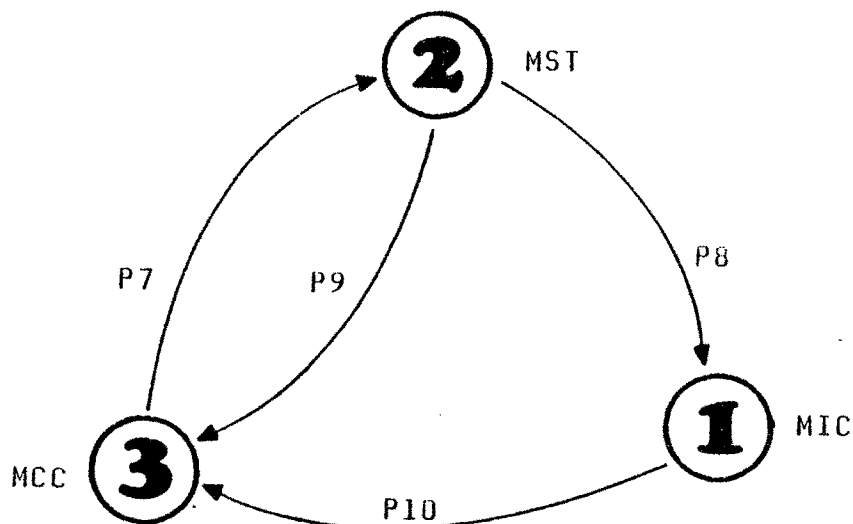


Fig. V-14 Automate des tampons du groupe C

Cet automate décrit la réception de paquets de données ou de

contrôle de la ligne série par le module MCC, leur analyse par le module MST et leur envoi au module MIC vers la Station Résiduelle avec retour des tampons vides au module MCC.

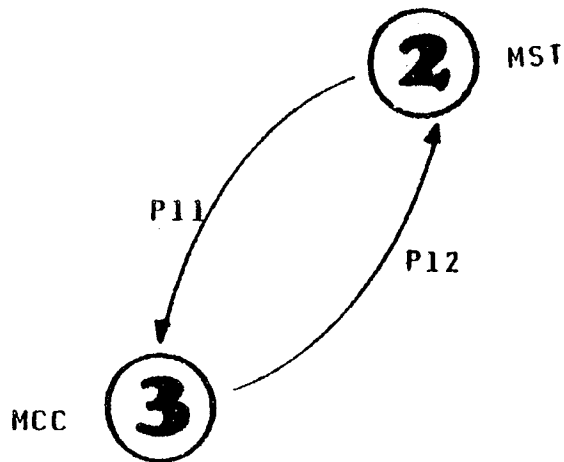


Fig. V-15 Automate des tampons du groupe D

Cet automate décrit le transfert des messages au module MCC par le module MST, avec retour des tampons vides au modules MST. Ce lien est utilisé par la Station Externe pour envoyer les messages de contrôle vers les Stations de Transport distantes.

#### V.2.2.4 Mécanisme d'Allocation

Afin de conserver la simplicité du mécanisme d'allocation des tampons et d'éviter l'utilisation des opérations indivisibles du type "test et modification" (test&set), nous l'avons implémenté à l'aide d'une seule variable partagé par tampon.

Le premier octet de chaque tampon est une variable ternaire (PROPT) qui indique à qui est alloué le tampon à un instant donné. Le mécanisme d'allocation est alors basé sur une seule règle :

Tous les modules ont droit d'accès en lecture à tous les tampons. Seule le module à qui le tampon est alloué à droit d'accéder en écriture.

Les différents modules peuvent être en attente active, en balayant les variables PROPT des tampons en lecture pour voir si l'un d'eux leur a été alloué par un autre module.

Sur un lien fonctionnel, la tâche consommatrice reste en attente tant que la tâche productrice (sur un autre module) ne le lui envoie (alloue) un tampon de mémoire. D'autre part, une tâche productrice ne fait pas de demandes explicites de tampons. Elle doit attendre qu'un tampon vide lui soit alloué (après avoir été consommé) par un autre module.

De cette façon aucun module ne peut s'allouer lui même un tampon, il doit attendre qu'un autre module le lui alloue. Un module ne peut allouer à un autre module seulement que l'un des tampons qui lui appartiennent.

L'intégrité du mécanisme d'allocation réside dans le fait que l'allocation d'un tampon à un module par le module propriétaire est faite en une seule opération d'écriture sur la variable PROPT correspondante. Une fois que le cycle d'écriture est fini, le tampon de mémoire est alloué à un autre module. D'autre part, l'allocateur du bus (unique) partagé assure l'indivisibilité de cette opération.

### V.3 - SYNCHRONISATION ENTRE MODULES

La synchronisation entre les tâches productrices et consommatrices se trouvant dans des modules différents est faite implicitement par l'algorithme d'allocation des tampons. Cette synchronisation et les opérations permises à un module à un instant donné peuvent être décrites à l'aide de chemins de synchronisation (path expressions) introduits par Campbell et Habermann (38).

#### V.3.1 Chemins de Synchronisation

La synchronisation des opérations exécutées par plusieurs processus sur un objet, est contenue dans la description de son type. La définition du type d'un objet inclut alors :

- la description des champs de données et leur valeur initiale,
- le chemin de synchronisation qui décrit l'ordre d'exécution des actions sur un objet,
- la définition des procédures et opérations possibles sur l'objet.

type nom

debut définitions des données

chemin..... fin

opérations définition des opérations

fin



L'expression de chemin est une expression régulière qui décrit les sequences d'exécution des actions sur un objet. Elle est définie à l'aide des opérateurs suivants :

- ; dénote séquençement,
- + dénote sélection exclusive,
- \* dénote répétition.

En utilisant cette notation, nous décrivons par la suite la définition des 4 groupes de tampons, ainsi que celles des processus associés exécutés par les modules.

Afin d'exprimer la synchronisation entre les différents modules nous utilisons un ensemble de procédures vides (P1 à P12). Physiquement l'exécution de ces procédures correspondent aux transferts des tampons de mémoire par le module propriétaire à un autre module (cf. Fig. V-12 à V-15).

### V.3.2 Le lien MIC-MST

La description suivante de type définit les tampons dédiés à la liaison du module interface canal MIC au le module station de transport MST (groupe A).

type Tampon-A

debut

chemin P1;((P2;P4)+P3) fin

chemin P1;Lire1 fin

chemin P2;lire2 fin

chemin Ecrire1;(P3+P4) fin

opérations

procédure P1 vide

procédure P2 vide

procédure P3 vide

procédure P4 vide

procédure Lire1 co lecture et analyse des données  
provenant de la Station  
Résiduelle co

procédure Lire2 co lecture des données et envoit  
sur la ligne série co

procédure Ecrire1 co remplissage du tampon à  
partir des données provenant  
du canal co

procédure Ecrire2 co ajoute entête pour CIGALE et  
pour la Station de Transport  
distante co

fin

Le lien, étant composé par un ensemble de 3 tampons, est décrit alors par :

LIEN-A : tableau (1:3) de Tampon-A

Les processus exécutés par le modules MIC, MST et MCC qui correspondent à la gestion de ce lien sont respectivement les suivants :

processus MIC-Lien-A

tantque données reçus du canal faire  
 Ecrire1  
 P1  
fait

processus MST-Lien-A

tantque vrai répéter  
 Lire1  
si données alors  
 Ecrire2  
 P2  
sinon  
 P3 co message de contrôle co  
fsi  
fait

processus MCC-Lien-A

tantque vrai répéter  
 Lire2  
 P4  
fait

### V.3.3 Le lien MSI-MIC

L'expression du type suivante définit les tampons dédiés à la liaison du module station de transport MSI au module interface canal MIC (groupe B).

type Tampon-B

debut

chemin P5;P6 fin

chemin Ecrire3;P6 fin

chemin P5;Lire3 fin

opérations

procédure P5 vide

procédure P6 vide

procédure Ecrire3 co envoi d'un message de  
contrôle vers la Station  
Résiduelle co

procédure Lire3 co lecture des données et envoit  
vers le canal co

fin

La description de ce lien, composé par 2 tampons, est alors :

Lien-B : tableau (1:2) de Tampon-B

Les processus qui gèrent ce lien et qui sont exécutés par les modules MST, et MIC, sont respectivement :

processus MST-Lien-B

tantque message à envoyer vers faire  
la Station Résiduelle

Ecrire3

P5

fait;

processus MIC-Lien-B

tantque vrai répéter

Lire3

P6

fait;

#### V.3.4 Le lien MCC-MST

L'expression de type suivante définit les tampons dédiés à la liaison du module contrôleur de communication MCC au module station de transport MST (groupe C).

##### type Tampon-C

###### debut

chemin P7;((P8;P10)+P9) fin

chemin P7;Lire4 fin

chemin P8;Lire5 fin

chemin Ecrire4;(P9+P10) fin

###### opérations

procédure P7 vide

procédure P8 vide

procédure P9 vide

procédure P10 vide

procédure Ecrire4 co remplissage du tampon à partir des données provenant de la ligne série co

procédure Lire4 co lecture et analyse des données provenant des stations distantes co

procédure Ecrire5 co ajoute information de contrôle pour la Station Résiduelle co

procédure Lire5 co lecture des données et envoi vers le canal co

###### fin

La description de ce lien, composé par 3 tampons, est alors :

Lien-C : tableau (1:3) de Tampon-C

Les processus associés a ce lien, exécutés par les modules MCC, MST, et MIC sont respectivement :

processus MCC-Lien-C

tantque données reçues de la ligne série faire

Ecrire4

P7 .

fait

processus MST-Lien-C

tantque vrai répéter

Lire4

si controle alors

P9

sinon

Ecrire5

P8

fsi

fait

processus MIC-Lien-C

tantque vrai répéter

Lire5

P10

fait

### V.3.5 Le Lien MST-MCC

La description de type fait par la suite, définit les tampons dédiés à la liaison du module station de transport MST au module contrôleur de communication MCC (groupe D).

type Tampon-D

debut

chemin P11;P12 fin

chemin Ecrire6;P12 fin

chemin P11;Lire6 fin

opérations

procédure P11 vide

procédure p12 vide

procédure Ecrire6 co envoi d'un message de  
contrôle vers une station  
distante co

procédure Lire6 co lecture des données et envoit  
sur la ligne série co

fin

La description de ce lien, composé par 2 tampons est la suivante :

Lien-D : tableau (1:2) de Tampon-D



Les processus qui gerent ce lien sont les suivants :

processus MST-Lien-D

tantque message de contrôle à envoyer faire  
vers une station distante

Ecrire6

P11

fait

processus MCC-Lien-D

tantque vrai répéter

Lire6

P12

fait

CHAPITRE VILES MODULES DU SYSTEME PIAR

- 1 - Le module Contrôleur de Communication
  - .1 Description fonctionnelle
    - .1 Gestion de la ligne physique
    - .2 Gestion de la ligne logique
    - .3 Gestion de l'interface avec les autres modules
  - .2 Description du matériel
  - .3 Description du logiciel
  
- 2 - Le module Station de Transport
  - .1 Description du matériel
  - .2 Structure de la Station Externe
    - .1 Le processus P-INTRES
    - .2 Le processus P-EMIS
    - .3 Le processus P-RECEP
    - .4 Le processus P-REVEIL
    - .5 Le processus MONITEUR
  
- 3 - Le module Interface Canal
  - .1 Introduction
  - .2 Caractéristiques de l'UCB
  - .3 Description fonctionnelle
    - .1 Description du matériel
      - .1 La partie contrôle
      - .2 La partie interface canal
      - .3 La partie interface dispositif
    - .2 Description du logiciel



## VI.1 - LE MODULE CONTRÔLEUR DE COMMUNICATION

Le module Contrôleur de Communication (MCC) prend en charge la gestion de la liaison du système PIAR (et indirectement de l'ordinateur hôte) avec le noeud de CIGALE. Cette liaison est faite au moyen d'une ligne série synchrone, géré selon la procédure IMM-UCIG en mode "full-duplex".

### VI.1.1 Description Fonctionnelle

Les fonctions exécutées par le MCC peuvent être décomposées en plusieurs niveaux :

- gestion de la ligne physique,
- gestion de la ligne logique,
- gestion de l'interface avec les autres modules

#### VI.1.1.1 Gestion de la ligne physique

La gestion de la ligne physique regroupe les fonctions assurées par ce qui est appelé couramment un coupleur de ligne série. Ce coupleur exécute les fonctions suivantes :

- la gestion des signaux de contrôle échangés avec le MODEM de la ligne en suivant les normes RS-232-C, avis V24 du CCITT,
- la conversion série/parallèle et parallèle/série des informations sur la ligne,
- la détection des caractères de synchronisation qui précèdent chaque message reçu de la ligne,
- la vérification en réception et la génération en émission du code cyclique redondant qui valide les messages de données.

#### VI.1.1.2 Gestion de la ligne logique

La gestion de la ligne logique regroupe les fonctions exécutées par la procédure de communication, dans ce cas TMM-UCIG. Ces fonctions assurent :

- la validité du format des messages,
- la validité de la syntaxe du contenu des messages,
- l'élimination en réception et adjontion en émission des caractères de contrôle prévus par la procédure,
- la détection et tentative de récupération des erreurs dûs à la perte ou à la duplication des messages.

#### VI.1.1.3 Gestion de l'interface avec les autres modules

L'interface avec les autres modules est faite au moyen de la mémoire commune. La gestion de cette interface est réalisé en exécutant l'algorithme d'allocation des tampons de mémoire.

Les traitements effectués par le MCC sont :

- émission sur la ligne du contenu d'un tampon de mémoire plein qui lui à été alloué,
- retour au propriétaire du tampon mémoire une fois que le noeud acquite la réception des données,
- transfert au module MST des tampons de mémoire pleins contenant les blocs de données qu'ont été reçus de la ligne sans erreur,
- envoie au module MST d'un message d'alarme si une erreur catastrophique est détecté (ligne coupé par exemple).

Les tâches du MCC produisent des tampons de mémoire du groupe C et consomment des tampons des groupes A et D (cf.V.3).

#### VI.1.2 Description du Matériel

Le module Contrôleur de Communication est bâti autour d'un microprocesseur Zilog Z80. La configuration du module est la suivante :

- un CPU Zilog Z80,
- un SIO Zilog,
- 2 Koctets de mémoire morte,
- 128 octets de mémoire vive,
- circuits d'interface V24,
- circuits de génération des signaux de contrôle,
- circuits de génération des temporisations.

Le circuit périphérique SIO exécute les fonctions de gestion de la ligne physique tandis que le CPU exécute les fonctions de gestion de la ligne logique (procédure de communication) et l'algorithme d'allocation des tampons de mémoire. Le schéma d'interconnexion est donné dans la figure VI-1.

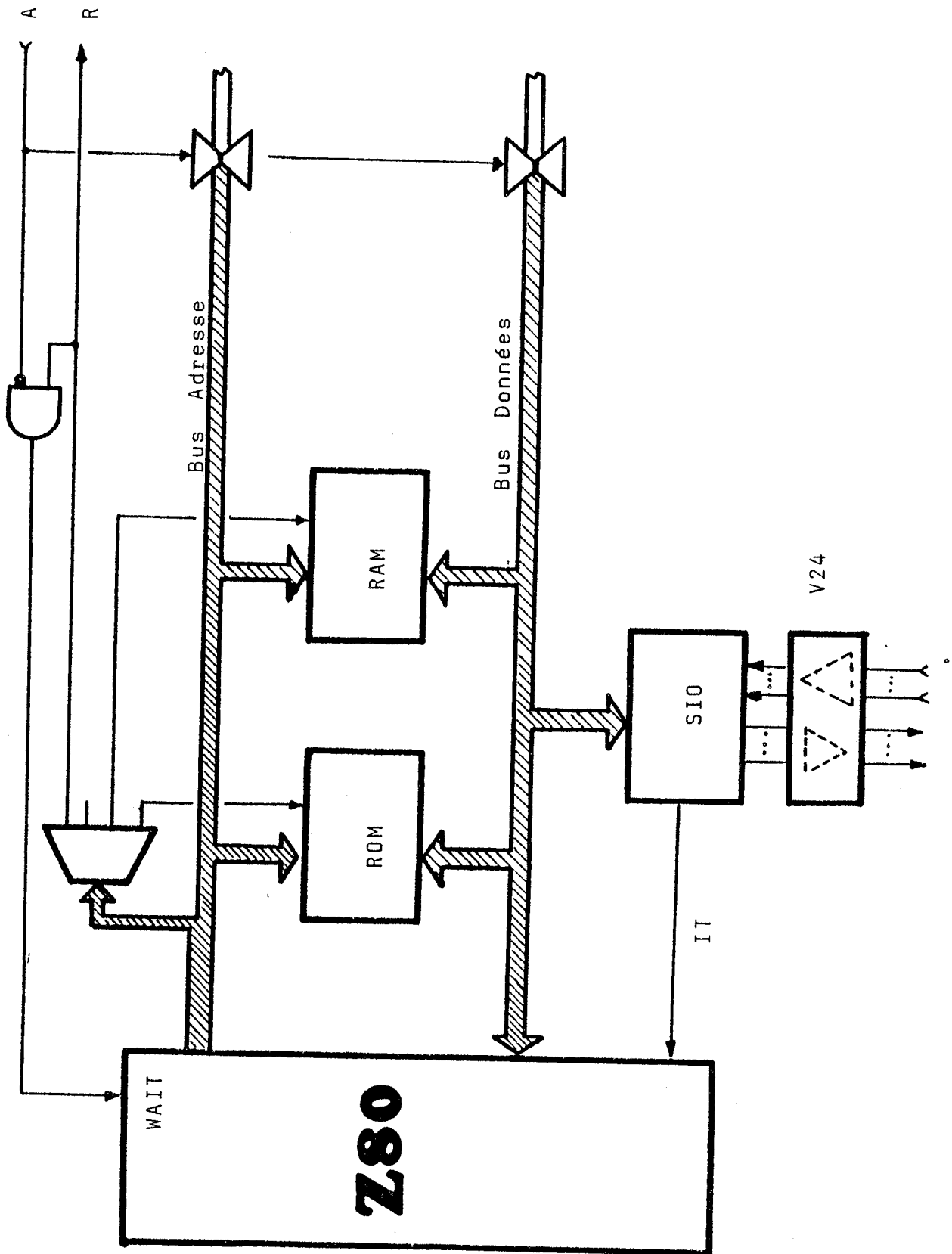


Fig. VI-1 Architecture interne du MCC

Le circuit SIO envoie ses requêtes de services au CPU sous forme d'interruptions. Nous avons utilisé pour cela la puissante structure d'interruptions vectorisées de ces circuits. L'adresse de la routine qui traite une interruption est formée par la concaténation du contenu du registre I du CPU, au contenu du registre IV du SIO et des 3 bits résultant de la combinaison du contenu du registre d'état interne du SIO.

De cette façon on minimise le temps de réponse à une interruption car il n'est pas nécessaire d'avoir un moniteur d'interruptions. La routine qui traite un type d'interruption est automatiquement activée quand ce type d'interruption apparaît.

### VI.1.3 Description du Logiciel

Le logiciel du MCC est décomposé en 7 processus : MONITEUR, INTMEM, RECEPTION, EMISSION, TRANCTL, TRANMESS, et RECMESS. Ces processus communiquent entre eux au moyen de deux vecteurs de bits, EVNTV et RQSTV. Dans ces vecteurs, chaque bit a une signification particulière et est utilisé par les processus pour positionner une requête ou indiquer un événement qu'ils ont produit ou pris en compte.

Le processus MONITEUR est en fait un allocateur du CPU car il fait la gestion des vecteurs EVNTV et RQSTV, et des événements externes, et active (d'une façon directe ou indirecte) les tâches correspondantes. MONITEUR comporte aussi les séquences d'initialisation du module. Il est activé lui même à l'initialisation et perd le contrôle seulement en faveur des processus qu'il active. Une fois que ces processus ont fini leur traitement ils rendent le contrôle au processus MONITEUR.



Le processus INTMEM fait la gestion de l'interface du module avec les autres modules en exécutant l'algorithme de l'allocateur de tampons de mémoire. Il exécute les procédures P4 (cf.V.3.2), P7 (cf.V.3.4) et P12 (cf.V.3.4). Il est activé par les événements suivants :

- fin d'émission et acquittement d'un bloc de données,
- réception d'un bloc de données de la ligne,
- détection d'une erreur catastrophique.

Le processus RECEPTION exécute l'algorithme de réception de la procédure de communication. Il est activé par la réception d'un message de données ou de certains messages de contrôle provenant de la ligne.

Le processus EMISSION exécute l'algorithme d'émission des données de la procédure de communication. Il est activé par l'allocation au MCC d'un tampon de mémoire plein à être envoyé sur la ligne et par la réception de certains messages de contrôle provenant de la ligne. Ce processus comporte la routine d'initialisation du dialogue avec le noeud.

Le processus TRANCTL envoie un message de contrôle sur la ligne, octet par octet. Il est activé indirectement quand il y a un message de contrôle à envoyer. Directement il est activé par les interruptions à chaque octet de l'émetteur du SIO.

Le processus TRANMESS envoie un message de données sur la ligne. Il ajoute les caractères de contrôle prévus par la procédure. Il est activé indirectement quand il y a un message de données à envoyer sur la ligne. Directement il est activé, octet par octet par les interruptions de l'émetteur du SIO. TRANMESS exécute les procédures Lire2 (cf.V.3.2) et Lire6 (cf.V.3.5).

Le processus RECMESS reçoit les messages provenant de la ligne. Il vérifie leur syntaxe et élimine les caractères de contrôle de la procédure. Il est activé indirectement à l'initialisation du module. Il est activé octet par octet, par le récepteur du SIO après que celui-ci a détecté le ou les caractères de synchronisation qui précèdent les messages reçus sur la ligne. RECMESS exécute la procédure Ecrire4 (cf.V.3.4).

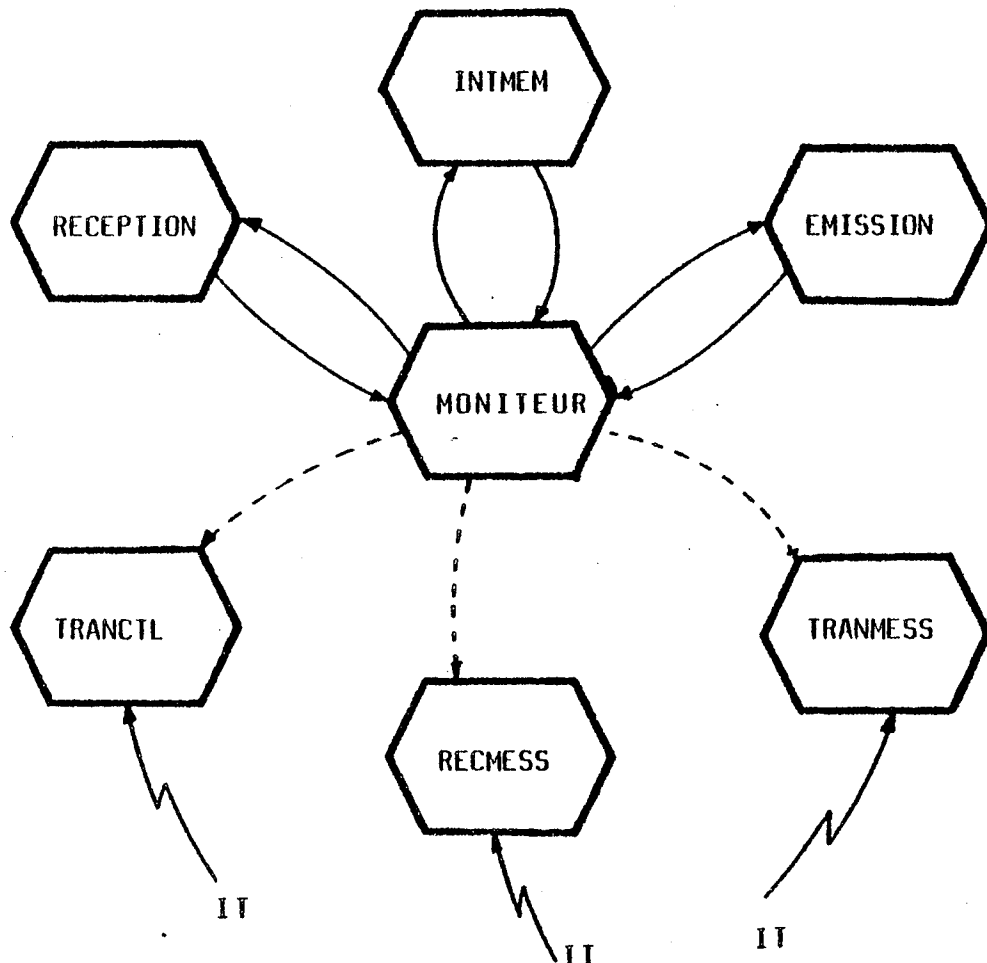


Fig. VI-2 Le logiciel du module MCC

## VI.2 - LE MODULE STATION DE TRANSPORT

Le module Station de Transport (MST) est le seul module du système PIAR qui n'a pas d'interface avec l'extérieur du système. Il exécute seulement des traitements internes au système et n'a donc pas de contraintes temporelles strictes dans ses traitements. Le MST exécute les fonctions que nous avons défini pour la Station Externe (cf.III.4) plus les fonctions d'interface avec les autres modules du système.

### VI.2.1 Description du Matériel

Le MST est bâti autour d'un microprocesseur Zilog Z80. La configuration du module est la suivante :

- un CPU Zilog Z80,
- un circuit compteur de temporisations CTC,
- 8 Koctets de mémoire morte,
- 4 Koctets de mémoire vive,
- circuits génération des signaux de contrôle.

Le circuit CTC, tient à jour l'heure locale du module et envoie une interruption au CPU tout les quanta de temps. La mémoire morte contient les programmes du module. La mémoire vive est utilisée pour les variables de travail et en particulier pour les tables qui décrivent les flots et leur activité.

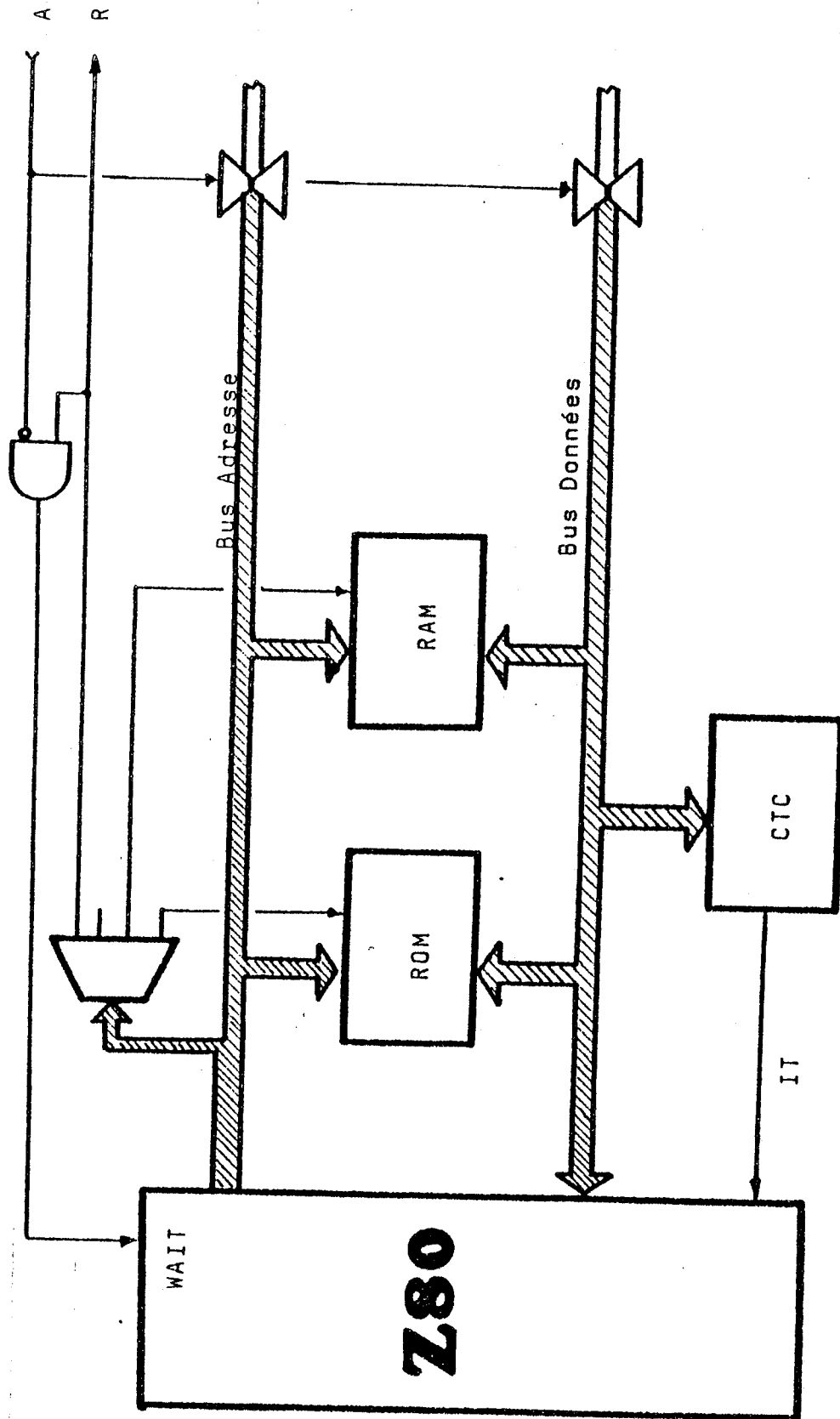


Fig. VI-3 Architecture interne du MST

### VI.2.2 Structure de la Station Externe

La structure que nous avons définie pour la Station Externe est inspirée de celle proposée pour le réseau CYCLADES (19) que nous avons présenté précédemment (cf. II.3). La Station Externe (SE) est divisée en 5 processus inter-reagissant entre eux : P-INTRES, P-EMIS, P-RECEP, P-REVEIL et MONITEUR.

#### VI.2.2.1 Le processus P-INTRES

Le processus P-INTRES gère l'interface entre la SE et la Station Résiduelle SR, implanté sur l'ordinateur hôte. Il reçoit les tampons de mémoire dont le contenu a été envoyé par la SR via le canal et le module MIC (cf.VI.3). Ce tampon peut contenir, soit un bloc de demande d'un abonné local, soit un bloc de données (fragment de Lettre ou télégramme).

P-INTRES traite immédiatement les blocs de demande BD et, si c'est le cas, génère le bloc de réponse BR correspondant et l'envoie vers la SR via le module MIC. par contre, il transfère au processus P-EMIS, les blocs de données pour qu'il les traite et les envoie vers le réseau.

P-INTRES est activé chaque fois qu'un tampon de mémoire du groupe A est alloué au module MST (P1). Ce tampon contient forcément un bloc qu'a été reçu du canal par le module MIC (cf.V.3). Il perd le contrôle en faveur du processus P-EMIS (bloc de données) ou, à la fin de son traitement, rend le contrôle au processus MONITEUR. P-INTRES exécute la procédure Lire1 (cf.V.3.2).

### VI.2.2.2 Le processus P-EMIS

P-EMIS se charge de la préparation des paquets devant être envoyées sur le réseau. Il appelle la procédure de sélection SELECTEUR qui choisit un flot parmi ceux qui sont prêts à émettre. Si le bloc à émettre est un fragment de Lettre, il envoie vers la SR la commande lui indiquant d'envoyer ce fragment.

A la réception du fragment (que lui transfère P-INTRES), P-EMIS ajoute l'entête avec l'information de contrôle correspondante et transfère le tampon de la mémoire vers le module MCC qui se charge de l'envoyer sur la ligne. Si le bloc à envoyer est une commande, ou un télégramme, P-EMIS le traite immédiatement, en ajoutant toujours l'entête CYCLADES.

P-EMIS est activé par P-INTRES ou par l'évènement "il y a au moins un flot prêt à émettre". Un fois qu'il a fini son traitement il rends le contrôle au processus MONITEUR. P-EMIS exécute les procédures Lire1 et Ecrire2 (cf.V.3.2), Ecrire3 (cf.V.3.3) et Ecrire6 (cf.V.3.5).

### VI.2.2.3 Le processus P-RECEP

P-RECEP traite les paquets reçus de la ligne qui ont été transférés au module MST par le module MCC. Il vérifie que le paquet est correct et le traite immédiatement. S'il s'agit d'un fragment d'une lettre, P-EMIS transfère le tampon de mémoire vers la SR via le module MIC et fait la mise à jour de la table de bits correspondante. Si la Lettre est complète, il commande à la SR de la délivrer à l'abonné destinataire.

P-RECEP est activé à chaque fois qu'un tampon de mémoire du groupe C est alloué au module MST. Ce tampon contient des informations qu'on été reçues de la ligne série par le module MCC. Une fois qu'il a fini son traitement, il rends le contrôle au processus MONITEUR. P-RECEP exécute les procédures LIRE4 et Ecrire5 (cf.V.3.4), Ecrire6 (cf.V.3.5) et Ecrire3 (cf.V.3.3).

#### VI.2.2.4 Le processus P-REVEIL

P-REVEIL fait la gestion de toutes les temporisations lancées par les différents processus de la Station Externe. Il balaie la table de temporisations et marque celles qui ont échoué. Il positionne ensuite les flots correspondants dans l'état "pret à émettre".

P-REVEIL est activé à chaque quantum de temps par l'interruption qu'envoie le circuit CTC. Une fois qu'il a fini de balayer la table de temporisations, il rends le contrôle au processus interrompu.

#### VI.2.2.5 Le processus MONITEUR

Le processus MONITEUR gère les événements externes et internes du module MST. Il effectue les traitements suivants :

- séquence d'initialisation du module MST,
- l'allocation des tampons de mémoire commune (vides ou pleins) au MST,
- la gestion de la file MST-MIC,
- la gestion de la file MST-MCC,
- détection de l'évènement "il y a au moins un flot prêt à émettre",

MONITEUR est activé à l'initialisation du système et perd le contrôle en faveur des processus qu'il active en fonction des événements qu'il détecte. Ces processus lui rendent la main une fois qu'ils ont fini leur traitement.

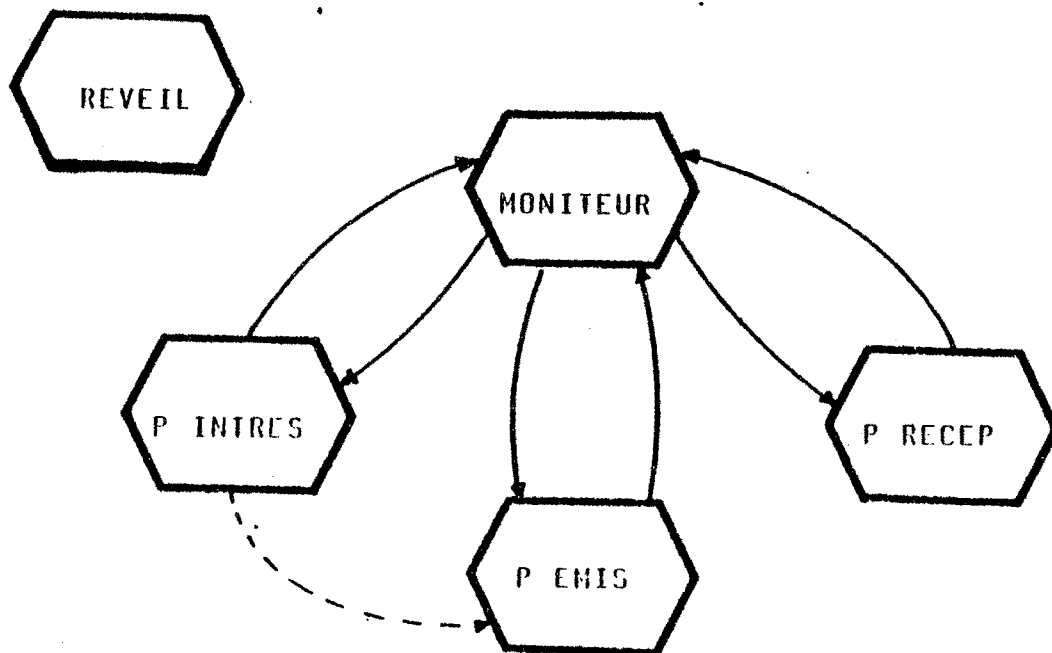


Fig. VI-4 Les processus du module MST



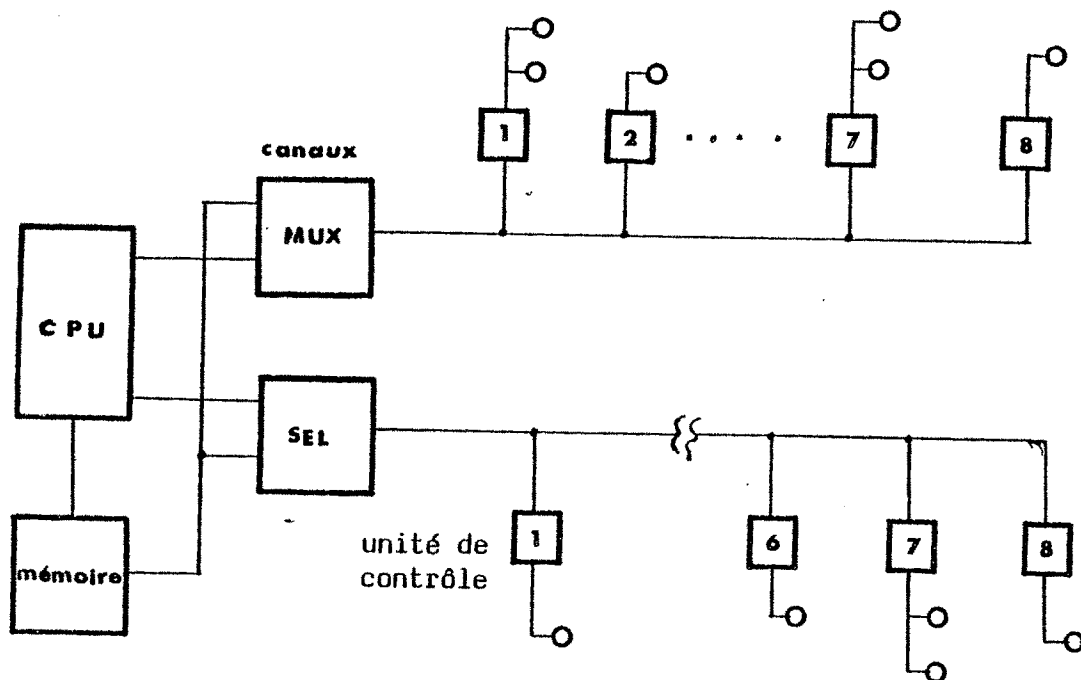
### VI.3 - LE MODULE INTERFACE CANAL

Le module Interface Canal (MIC), gère la liaison du système PIAR avec l'ordinateur hôte. Cette liaison est faite sur le bus canal du processeur d'entrées/sorties. Ce module est bati autour de l'Unité de Contrôle Banalisé (UCB) conçue et réalisée dans un projet conjoint avec M. Arnaud. La partie qui suit a été rédigée en commun et représente une partie de la thèse de M. Arnaud (25).

#### VI.3.1 Introduction

L'unité de Contrôle Banalisée (UCB) permet la liaison d'un dispositif quelconque à un ordinateur IBM de la gamme 360/370/303x. Pour cela, il est connecté à une des unités d'échange (ou canal) de cet ordinateur et est vu par ce canal comme une unité de contrôle.

Un système basé sur un ordinateur des gammes mentionnées se présente suivant la configuration simplifiée suivante :



Il existe deux types d'unités d'échange :

- les canaux simples (ou sélecteurs) qui opèrent en mode de transfert de type "burst" (c.a.d. qu'un seul dialogue a lieu à la fois) avec des unités de contrôle de dispositifs rapides (disques, tambours, etc...).
- les canaux multiples (ou multiplexeurs) qui travaillent indifféremment en mode "burst" ou en mode "multiplex". Dans ce cas, plusieurs transferts imbriqués ont lieu, octet par octet (par exemple), avec plusieurs dispositifs. Ce sera toujours le cas pour des unités gérant des périphériques lents (imprimantes, lecteurs/perforateurs de cartes, lignes de transmission etc....).

Le canal impose en outre à l'unité de contrôle certaines séquences de signaux à l'aide de lignes de contrôle (tag-lines) afin d'assurer correctement l'initialisation et le déroulement des échanges des données, et des informations de contrôle, d'analyse, d'états.

Les lignes de contrôle et les bus de données, ainsi que les circuits et/ou les programmes interprétant correctement ces séquences définissent l'Interface d'Entrée/sortie du système.

Une unité de contrôle peut :

- soit être identifiée à un dispositif d'entrée/sortie,
- soit gérer plusieurs dispositifs du même type.

Dans ce dernier cas, elle doit adapter les caractéristiques propres à chaque dispositif à celles de l'interface d'entrée/sortie.

Dans notre cas, il s'agira d'une réalisation mixte (architecture adaptée + programme en mémoire morte) autour d'un microprocesseur assurant également la gestion de l'interface avec le dispositif - de type quelconque a priori - connecté à l'ordinateur.

### VI.3.2 Caractéristiques de l'UCB

Cette unité de contrôle banalisée ainsi définie :

- se connecte à l'interface d'entrée/sortie des canaux simples ou multiples,
- est compatible avec les spécifications de cette interface définies par constructeur.
- peut adapter son mode d'opération de façon telle que la rapidité de l'appareil soit utilisée au mieux (le maximum d'octets transférés à la fois pendant la durée de la sélection imposée par le canal) tout en respectant la définition du mode multiple (plusieurs transferts imbriqués) afin de ne pas saturer le canal.
- permet une gestion programmée du dispositif connecté, donc elle est très souple pour s'adapter aux différents types de dispositifs.
- offre un débit maximum d'informations entre le dispositif et le canal de 400 Koctets/sec.
- ne nécessite qu'une seule tension d'alimentation (5V, 2A) et est contenue sur une carte standard 20x25 cm contenant une quarantaine de circuits intégrés plus quelques éléments discrets.

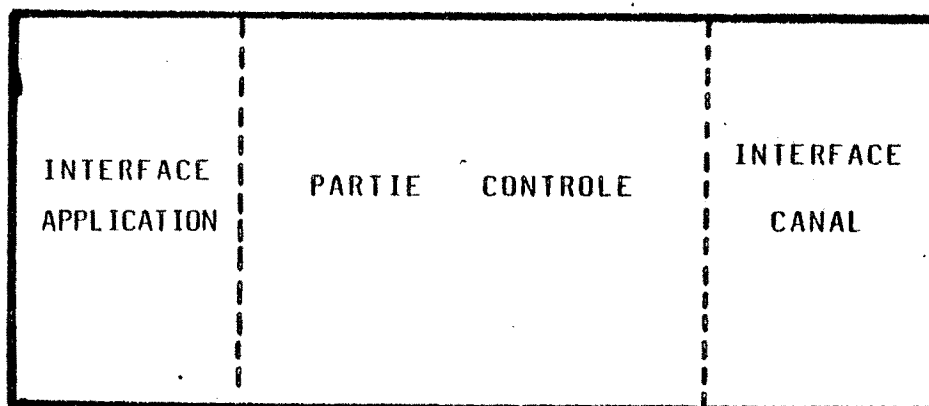
### VI.3.3 Description Fonctionnelle

Le module MIC exécute les fonctions associées à l'interface du système PIAR avec l'ordinateur hôte et les fonctions de l'interface avec les autres modules du système. Vis-à-vis de la mémoire commune, les tâches qu'il exécute produisent dans les tampons du groupe A et consomment des tampons des groupes B et C.

#### VI.3.3.1 Description du Matériel

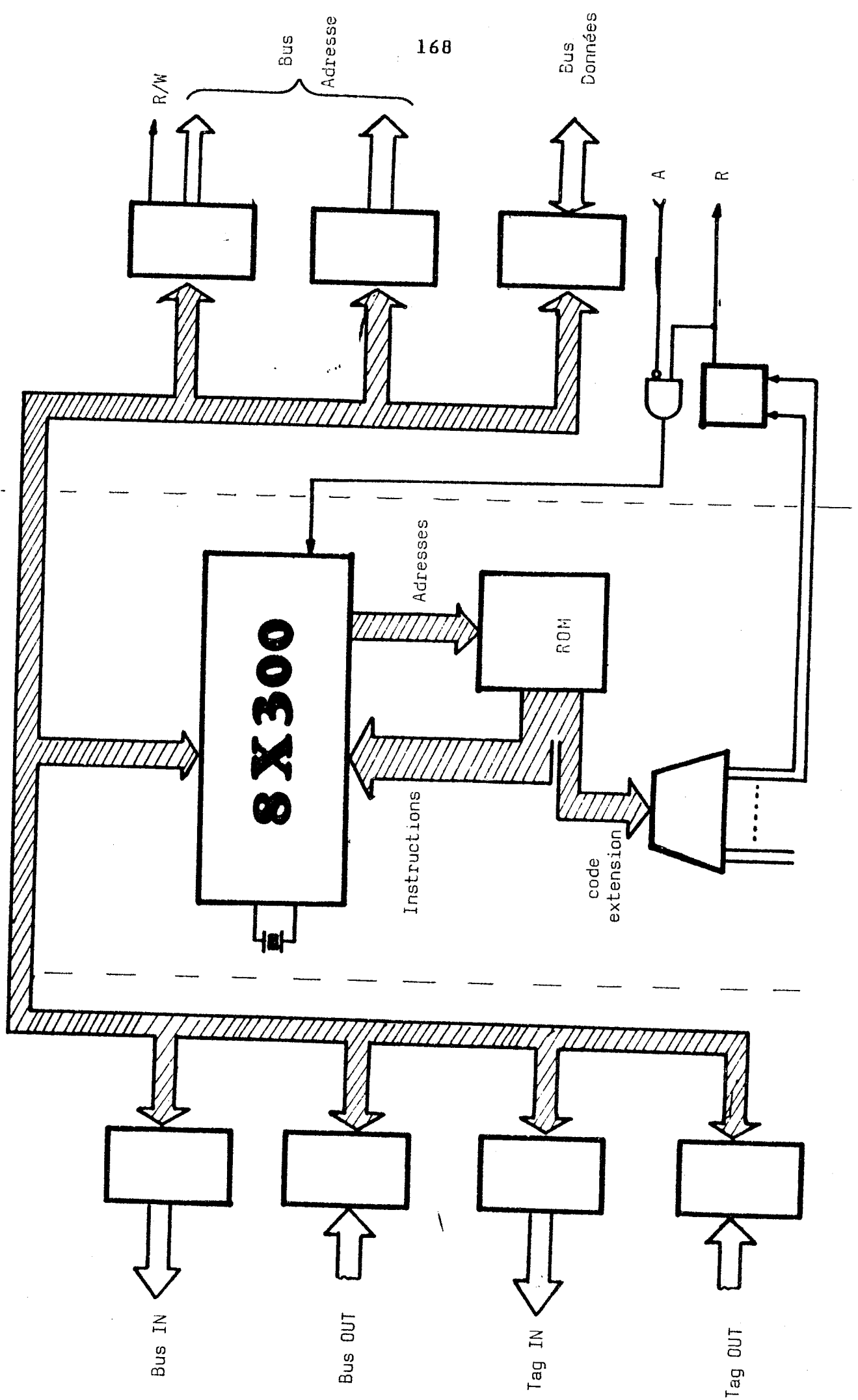
Du point de vue matériel, l'Unité de Contrôle Banalisée se décompose en trois parties :

- la partie contrôle,
- la partie interface canal,
- la partie interface dispositif.



Ces trois parties sont construites autour d'un bus interne commun ce qui conduit à l'architecture simplifiée indiquée dans la figure VI-6.

Fig. VI-5 Architecture interne de l'UCB



### VI.3.3.1.1 La partie contrôle

La partie contrôle assure le traitement de toutes les informations :

- séquençement des états des lignes de contrôle,
- transit des données,
- mise à jour des états internes et de ceux du dispositif,

La partie contrôle se décompose en 5 blocs fonctionnels :

- 1 - Un microprocesseur Signetics 8X300 ayant les caractéristiques suivantes :
  - . fréquence d'horloge jusqu'à 8 Mhz,
  - . temps de cycle d'instruction 250 nanosecondes,
  - . instructions sur 16 bits,
  - . capacité de mémoire programme 8 K mots,
  - . bus de données bi-directionnel de 8 bits,
  - . jeu de 8 instructions.
- 2 - Extension du jeu d'instructions sur 10 bits qui sont décodés par le matériel afin de générer des actions parallèles au déroulement des instructions.
- 3 - Mémoire vive de travail de 16 mots de 8 bits extensible à 256 mots.
- 4 - Mémoire morte de programme de 512 mots de 26 bits.
- 5 - Circuit de génération des états internes.

### VI.3.3.1.2 La partie Interface Canal

La partie d'interface avec le bus du canal assure la génération et la réception des lignes de contrôle et de données du canal en respectant les restrictions temporelles et électriques imposées par le constructeur (24). La liaison avec le canal est faite par 4 ensembles de lignes groupées de manière fonctionnelle :

- génération des "Tag-in", 8 lignes de sortie;
- génération de "Bus-in", 9 lignes de sortie;
- réception des "Tag-out", 8 lignes d'entrée;
- réception de "Bus-out", 9 lignes d'entrée.

La partie interface canal comporte aussi le circuit de reconnaissance automatique de la ou des adresses attribuées au dispositif ainsi connecté, choisies parmi les 256 adresses possibles. Ceci permet d'établir ou non la liaison logique avec le canal en bloquant la propagation de la ligne de sélection.

### VI.3.3.1.3 La partie Interface Dispositif

L'interface avec le dispositif est faite à l'aide des portes bi-directionnelles de 8 bits chacune. Ces portes permettent de mémoriser les signaux d'entrée ou de sortie vers le dispositif.

Dans le cadre du système PIAR, l'interface vers le dispositif est en fait l'interface vers la mémoire commune qui sert à faire communiquer les modules. Pour réaliser cette interface nous avons rencontré certaines difficultés :

- Le microprocesseur 8X300 n'a pas une architecture "classique". En effet, l'espace programme n'est pas banalisé avec l'espace des données. De ce fait, ce microprocesseur ne dispose pas d'un bus d'adresses pour l'espace données.

- Le cycle du microprocesseur 8X300 est beaucoup plus rapide que ceux des autres microprocesseurs du système. La durée des signaux (environ 60 nano-secondes) et en particulier des données n'est pas compatible avec le temps d'accès de la mémoire commune.

Pour faire l'interface avec la mémoire commune, nous avons construit avec des portes, le bus de données, le bus des adresses et la ligne de lecture/écriture vers la mémoire commune. Nous avons fabriqué aussi un circuit de synchronisation rapide permettant de dialoguer avec l'allocateur de bus au moyen des deux variables décrites précédemment (cf. V.2.1.2).

#### VI.3.3.2 Description du Logiciel

Le logiciel du module Interface Canal est divisé en trois processus: INTCAN, INTMEM et MONITEUR.

INTCAN gère l'interface avec le canal. Il exécute les algorithmes de poignée de main pour chacun des échanges avec le canal. Il est activé par une sélection provenant du canal ou par l'allocation d'un tampon de mémoire commune avec de données à envoyer au canal. INTCAN fait les traitements suivants :

- réception et décodage des commandes d'entrée/sortie.
- présentation des états en attente,
- transferts des données,
- test de fin de "burst",
- génération d'interruption d'entrée/sortie à l'unité centrale par présentation au canal d'un état asynchrone.
- détection des séquences spéciales,
  - .déconnexion de l'interface,
  - .chainage des commandes,
  - .suppression des échanges.



INTMEM exécute l'algorithme d'allocation des tampons de mémoire et synchronise donc les échanges avec les autres modules du système PIAR (cf. V.3) Il est activé soit par un bloc de données qui a été reçu du canal, dans ce cas il transfère le tampon correspondant au module Station de Transport; soit par la fin d'un transfert de données vers le canal, dans ce cas il rend le tampon vide au module propriétaire.

MONITEUR fait l'initialisation du module et assure la gestion des évènements externes au module :

- sélection de la part du canal,
- allocation d'un tampon de mémoire plein,
- allocation d'un tampon de mémoire vide.
- initialisation (générale ou sélective) de la part du canal.

Ce processus est activé à l'initialisation et active à son tour les processus INTCAN et INTMEM au fur et à mesure que les évènements externes arrivent. Ces processus lui rendent la main une fois qu'ils ont fini le traitement correspondant.

CONCLUSION



## CONCLUSION

Le Périphérique Intelligent d'Accès au Réseau que nous avons conçu, permet le raccordement d'un ordinateur de la gamme IBM 360/370/303x au réseau CYCLADES. Il supporte la plupart du logiciel d'interface au réseau et, en particulier, la gestion de tous les évènements en temps réel. Il reste en charge à l'ordinateur hôte le logiciel d'interface entre les abonnés locaux et le système PIAR, comme la méthode d'accès d'un périphérique quelconque.

La station de transport se trouve donc implanté dehors du système hôte, sur un ordinateur frontal de communication. Le système PIAR se présente comme une alternative avantageuse face à l'implantation de la SI sur l'ordinateur hôte lui même ou son implantation sur un frontal basé sur un mini-ordinateur ou un processeur de communication spécialisé tel que un IBM 3870 ou un HB-IOM.

La modularité de l'architecture logicielle et matérielle du système PIAR nous a permis de développer et mettre au point chacun des modules séparément. De même, cette modularité laisse envisager ses possibilités d'expansion. En effet, il est possible de connecter le système (et indirectement l'ordinateur hôte) à deux (ou plus) noeuds du réseau de communication CIGALE en ajoutant un autre module Contrôleur de Communication MCC. Dans ce cas, le module Station de Transport MST, exécute aussi les fonctions de routage des messages vers un ou autre noeud. Cette configuration est très intéressante car la fiabilité de la connexion de l'ordinateur hôte au réseau est accrue en particulier dans le cas où un des noeuds tombe en panne.

Par rapport à l'implémentation actuelle de la ST sur l'IBM 360/67 du CICG, le système PIAR offre une sécurité accrue car les abonnés locaux sont complètement séparés de la ST. Une erreur d'un abonné ne peut pas avoir de conséquences pour la ST (38). De plus, l'ordinateur hôte se trouve déchargé de tout le travail en temps réel (gestion de la ligne et temporisations de la ST) ainsi que d'une partie importante du service de transport. La ligne est gérée avec une procédure full-duplex, plus efficace que la procédure BSC.

Une critique qu'on peut faire au système PIAR est sa spécificité vis à vis de la connexion à l'ordinateur hôte et de la procédure de communication utilisée vers le noeud. Cependant, grâce à la modularité du système, si on veut l'adapter à d'autres environnements, il suffit de changer le module d'interface correspondant. En particulier, le matériel du module MCC, ne réquiert aucun changement pour l'adapter aux nouvelles procédures orientées au bit tel que HDLC et en particulier aux normes X25 du CCITT. Seulement le logiciel doit être modifié, et ceci sans grandes problèmes.

REFERENCES



REFERENCES

- (1) Frank H. and W. Chou, Topological Optimisation of Computer Networks  
National Computer Conference, 1975.
- (2) Wilkov R., Analysis and Designs of Reliable Computer Networks,  
IEEE Transaction on Communications Juin 1972.
- (3) The CYCLADES Network Present State and Developpement Trends,  
Réseau CYCLADES, RES 505-1, Mars 1975.
- (4) IRIA, Presentation du Réseau CYCLADES,  
Novembre 1973.
- (5) IRIA, Le Réseau CYCLADES, annuaire des ressources disponibles,  
Novembre 1973.
- (6) Seguin J., Traitement Distribué d'Informations Reparties dans les Réseaux d'Ordinateurs,  
Thèse Docteur d'Etat USMG, Grenoble Mars 1978.
- (7) Sergeant G., Farza M., Machine Interpretative pour mise en oeuvre d'un langage de commande sur le réseau CYCLADES,  
Thèse 3ème Cycle, Toulouse, octobre 1974.



- (8) Naffah N., Etude de la Gestion des Terminaux dans un Réseau Général Informatique et Développement d'un système Microprogrammé pour la connexion directe d'un terminal intelligent sur le Réseau CYCLADES.  
Thèse Docteur Ingenieur, Université Paris VI, décembre 1975.
- (9) Crocker S. et Al., Function Oriented Protocols for the ARPA Computer Network,  
AFIPS Conference Proceedings, Vol. 40, 1972.
- (10) Pouzin L., CIGALE, The packet switching machine of the CYCLADES computer network.  
Réseau CYCLADES MIT-556, Novembre 1973.
- (11) Dromard D., Répresentation d'une Procédure de Transmission par des Automates d'Etats Finis,  
Thèse Docteur Ingenieur, Université Paris VI, Juin 1974.
- (12) Laroche J.P., Naffah N., Specifications d'interface TMM avec le Réseau CIGALE,  
Réseau CYCLADES, TRA-521, Décembre 1975.
- (13) Gelenbe E. et Al., Performance Limits of the TMM protocol: Modelling and Measurement.  
IRIA, Rapport de Recherche N. 230, Avril 1977.
- (14) Payne R.C., A Brief Description of CIGALE's MVB,  
CCNG, Waterloo, Canada, Août 1975.
- (15) Zimmerman H. Elie M., Transport Protocol Standard end-to-end Protocol for Heterogeneous Computer Networks.  
Réseau CYCLADES, SCH-519.2 mai 1975.

- (16) Pouzin L., CIGALE, The Packet Switching Machine of the CYCLADES Computer Network, Réseau CYCLADES, MIT-556, Novembre 1973.
- (17) Zimmerman H., Insertion d'une Station de Transport dans un Système d'Exploitation, Réseau CYCLADES, SCH-546, Janvier 1975
- (18) Chambon J.F. et Le Bihan B., Implantation de la Station de Transport dans un Calculateur Frontal, Ecole de Mines de St. Etienne, Rapport D.R.M.E., Juillet 1975
- (19) Garcia et al., Spécifications de Réalisation de la Station de Transport ST2 Portable, Réseau CYCLADES SCH-536.2 Mai 1975.
- (20) Papachristodoulou Z., TELCOM Un support de terminaux lourds sous un système à temps partagé, Thèse USMG, Octobre 1974.
- (21) Ng.X Dang et al, Presentation et définition de SYNCOP, IMAG, Rapport de recherche No. 64.
- (22) Ng.X Dang et al, SYNCOP Implémentation sous CP/67 IMAG, Septembre 1975.
- (23) Saettone R., On the decentralisation of a Transport Station of the CYCLADES network, IMAG, rapport DEA Juin 1976.
- (24) IBM, IBM System/360 and System/370 I/O Interface Channel to Control Unit Original Equipment Manufacturer Information Manual, GA22-6974-1, Juillet 1972.

- (25) Arnaud S., Conception et réalisation d'une Unité de Contrôle de Périphérique Banalisée, Thèse Docteur-ingénieur, INPG (à soutenir).
- (26) CCITT Recommended Standard RS-232-C  
Aôut 1969.
- (27) Weissenberg A.J., Analysis of Multiple Micro-processor System Architectures, Computer Design, Juin 1977
- (28) Anderson G.A. et Jensen E.D., Computer interconnection Structures : Taxonomy, Characteristics and Examples, Computer Surveys, Vol 7, No. 4, Décembre 1975.
- (29) Lillen H., Du microprocesseur au micro-ordinateur, Editions Radio, 1977.
- (30) Anceau F., Principes et mise en oeuvre des microprocesseurs, ENSIMAG, Janvier 1978.
- (31) Grenelle F., Le Multi Micro-ordinateur MICRAL M, R2E
- (32) Pechoucek M., Anomalous reponse times of input synchronisers, IEEE Transactions on Computers, Vol C-25, No. 2, Fevrier 1976.
- (33) Mayne D. et Moore R., Minimize computer craches, Electronic Design, Vol. 9, Avril 1974.
- (34) Quint V., Protection logicielle contre les erreurs dans un réseau d'ordinateurs heterogene. Application au 360/67 du réseau CYCLADES, Thèse Docteur-ingénieur INPG, Décembre 1976.

- (35) CCITT Study Group VII, Proposal for revised draft recommendation X25,  
No. 62 E, Novembre 1975.
- (36) Flynn M., Some computer organisations and their effectiveness,  
IEEE Transactions on Computers, Vol. C-21, No. 9,  
Septembre 1972.
- (37) Enslow P., Multiprocessor Organization. A Survey.  
Computing Surveys, Vol. 9. No. 1, Mars 1977.
- (38) Campbell et Habermann, The specification of process synchronisation by path expressions, Lecture Notes on Computer Science, Vol. 16, 1974.
- (39) Anceau F., Mecanismes Primitifs de Synchronisation au niveau Matériel, Symposium de Systèmes d'exploitation IRIA, France, octobre 1978.
- (40) Saettone R., MITS Microprocessor Implementation of a Transport Station,  
Computer Network Symposium, Liège, fevrier 1978.

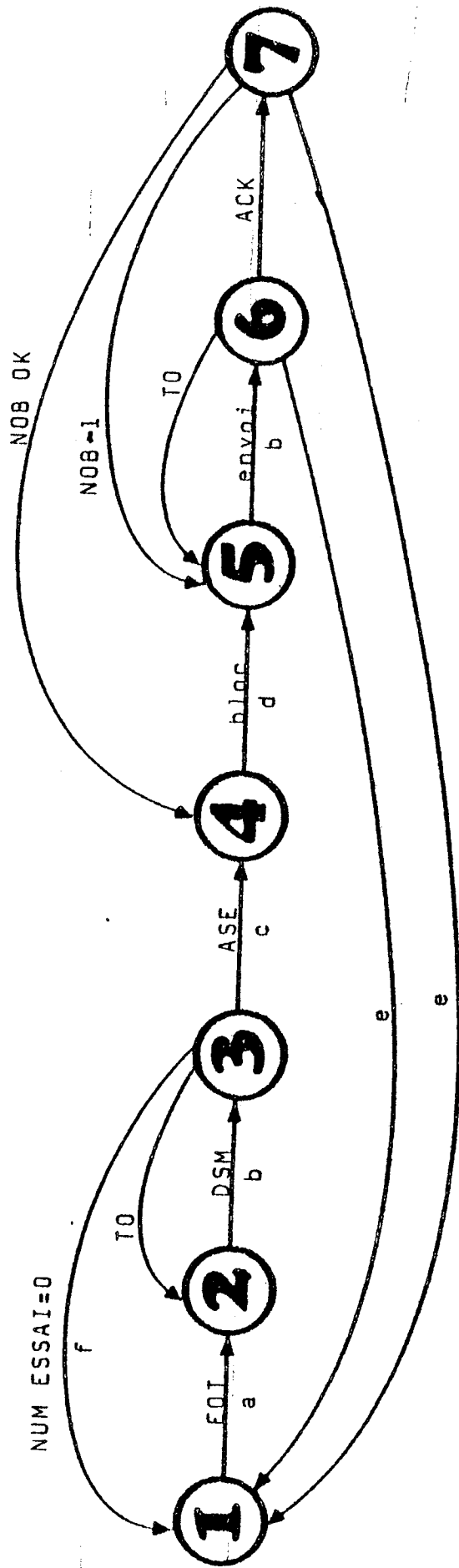


ANNEXE I

LOGICIEL DU MODULE CONTROLER DE COMMUNICATION



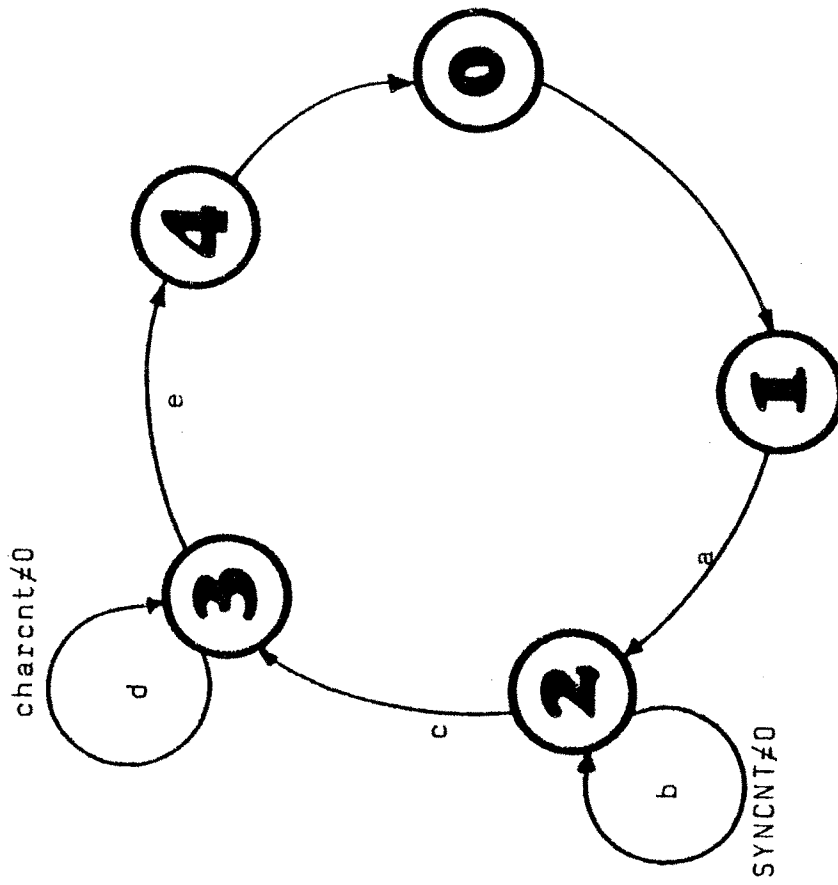
AUTOMATE D'EMISSION DE LA PROCEDURE TMM UCIG



- a NUM ESSAI = MAX1
- b NUM ESSAI = NUM ESSAI - 1
- c NOB = 0, ligne "up"
- d NUM ESSAI = MAX2, NOB = NOB + 1
- e ligne "down"
- f envoit message erreur au MST

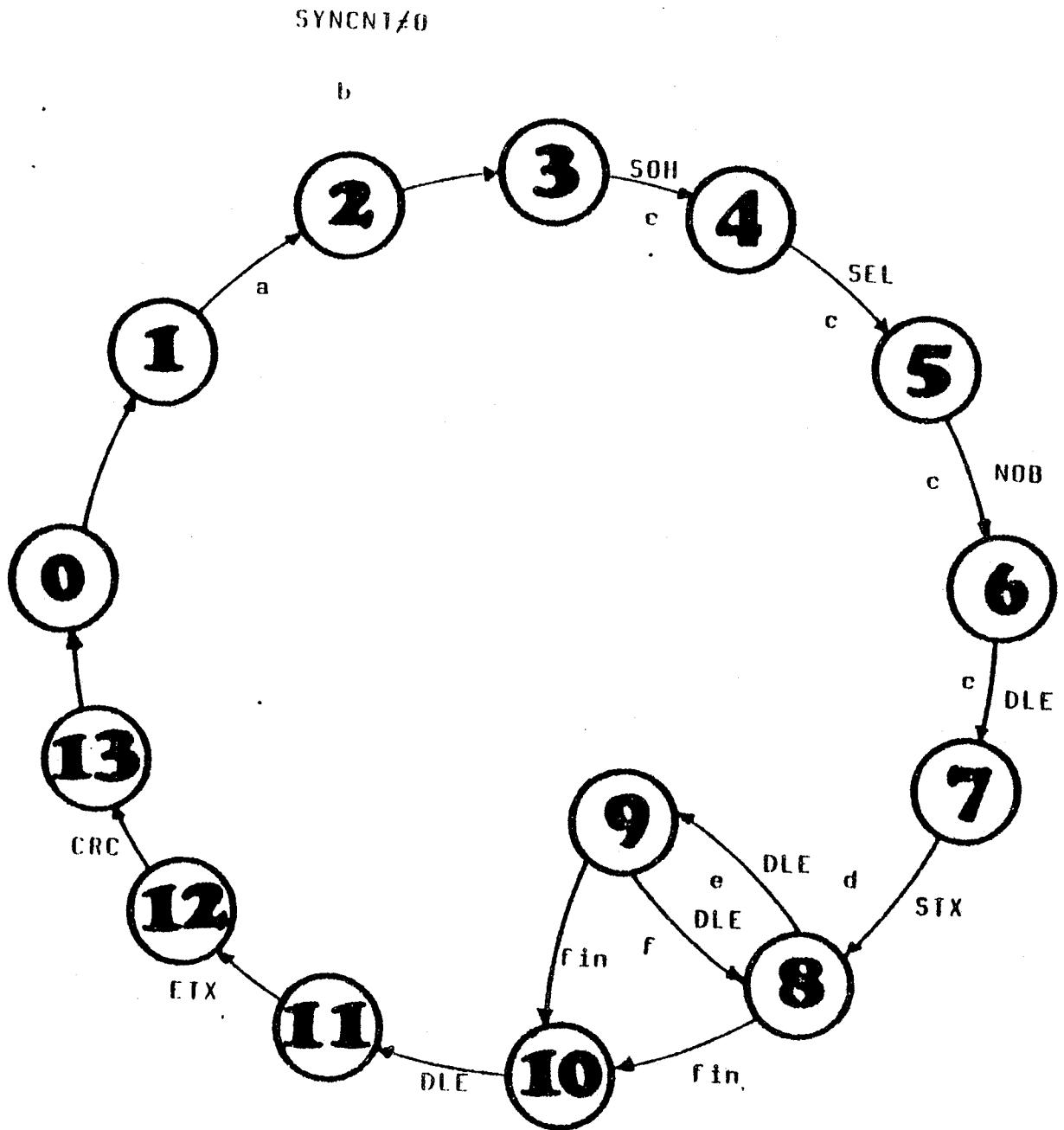


AUTOMATE D'EMISSION D'UN MESSAGE DE CONTROLE



- a : SYNCNT = 4
- b : SYNCNT = SYNCNT - 1, envoit SYN
- c : CHARCNT = 4
- d : CHARCNT = CHARCNT - 1, envoit CHAR
- e : inhiber emetteur

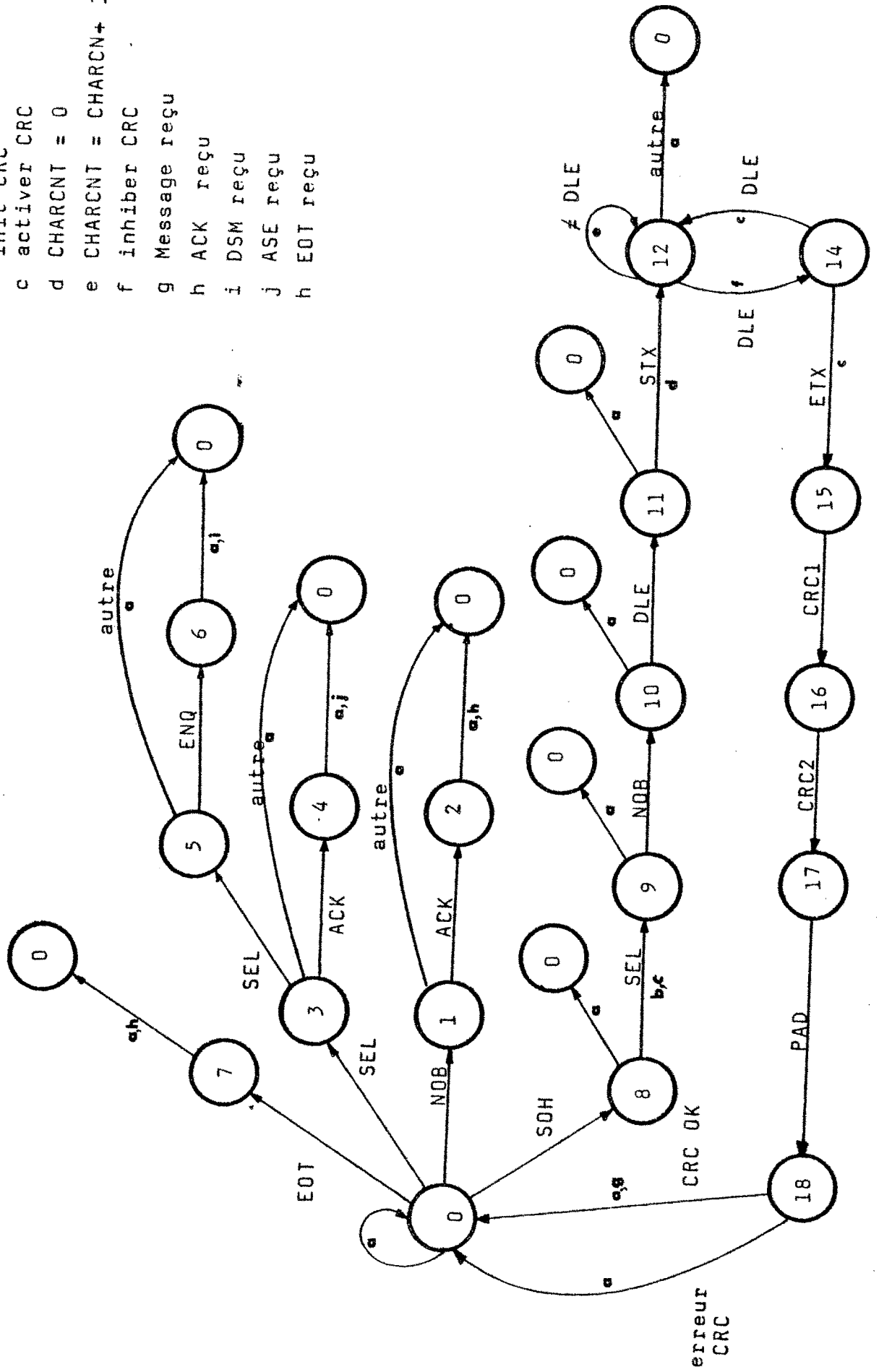
## AUTOMATE D'EMISSION D'UN MESSAGE - TRANSMISS



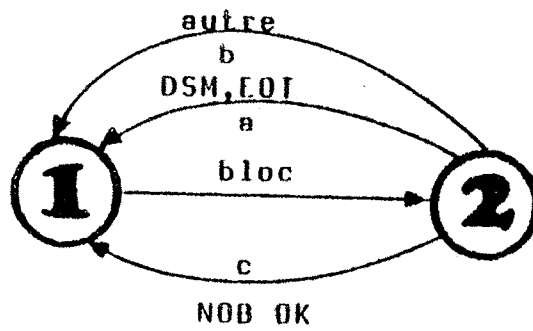
- a : SYNCNT = 4
- b : SYNCNT = SYNCNT - 1, envoie SYN
- c : envoie char
- d : reset CRC
- e : inhibe calcul CRC
- f : active CRC

AUTOMATE DE RECEPTION RECMESS

- a init Récepteur
- b init CRC
- c activer CRC
- d CHARCNT = 0
- e CHARCNT = CHARCNT + 1
- f inhiber CRC
- g Message reçu
- h ACK reçu
- i DSM reçu
- j ASE reçu
- h EOT reçu



## AUTOMATE DE RECEPTION DE LA PROCEDURE TMM UCIG



- a NOB = 0, envoie ASE
- b. envoie séquence précédente
- c NOB = NOB + 1, envoie ACK



ANNEXE II

LOGICIEL DU MODULE STATION DE TRANSPORT



Processus P-INTRES co activé par un tampon de mémoire du  
groupe A alloué au MST co

debut

si ENVSETG ou ENVSEFR  
alors activer P-EMIS

si ENVSEBD

alors

LIRE1

<traitement du bloc de demande>

P3

sinon

Proc-REPAB (bloc erroné)

P3

fin



Processus P-EMIS co activé par P-INTRES à la réception  
d'un bloc ENVSEFR ou ENVSETG, ou  
activé par MONITEUR quand il y a  
au moins un flot prêt à emettre co

debut

si ENVSEFR

alors

LIRE1

Proc-Verif-ID (identification de flot)

si identification erronée

alors

Proc-REPAB (identification de flot erronée)

sinon

fabriquer entête

fabriquer ACK

fabriquer CREDIT

m.a.j. DLTE co descripteur Lettre en émission co

si dernier fragment

alors

positionner bit EOL

activer temporisation attente ACK

ECRIRE2

P2

si ENVSETG

alors

LIRE1

fabriquer entête

ECRIRE2

P2

si flot pret-à-émettre

alors

si commande

alors

fabriquer entête

fabriquer commande

ECRIRE6

P11

si fragment

alors

fabriquer DEMSRFR

ECRIRE3

P5

fin

processus P-RECEP co activé par un tampon du groupe C  
alloué au MST co

debut

LIRE4

Proc-Verif-ID (identification de flot)

si identification inconnue

alors

Proc-Verif-ID (identification de porte)

si identification OK et FL-INIT

alors

Proc-Négociation (init)

P9

sinon

abandonne message

P9

sinon

si FL-TERM

alors

si service de base

alors

abandonne message

P9

sinon

Proc-Négociation (fin)

sinon

si Télégramme

alors

si télégrammes acceptés

alors

fabriquer ENVSRTG

Ecrire5

P8

sinon

abandonne message

P9

si ACK ou CRD

alors

<traitement de ACK-CRD>

P9

si Fragment de Lettre

alors

<traitement de Fragment>

fabriquer ENVSRFR

Ecrire5

P8

sinon

abandonne message

P9

fin

processus P-REVEIL co activé par interruption généré par  
le circuit de temporisations co

debut

tantque fin-de-file = faux faire

<traitement de bloc-réveil>

si temporisation échoué

alors positionner flot "pret-à-émettre"  
progression dans la file

fait

fin

ANNEXE III

LOGICIEL DU MODULE INTERFACE CANAL

travail conjoint avec M. Arnaud

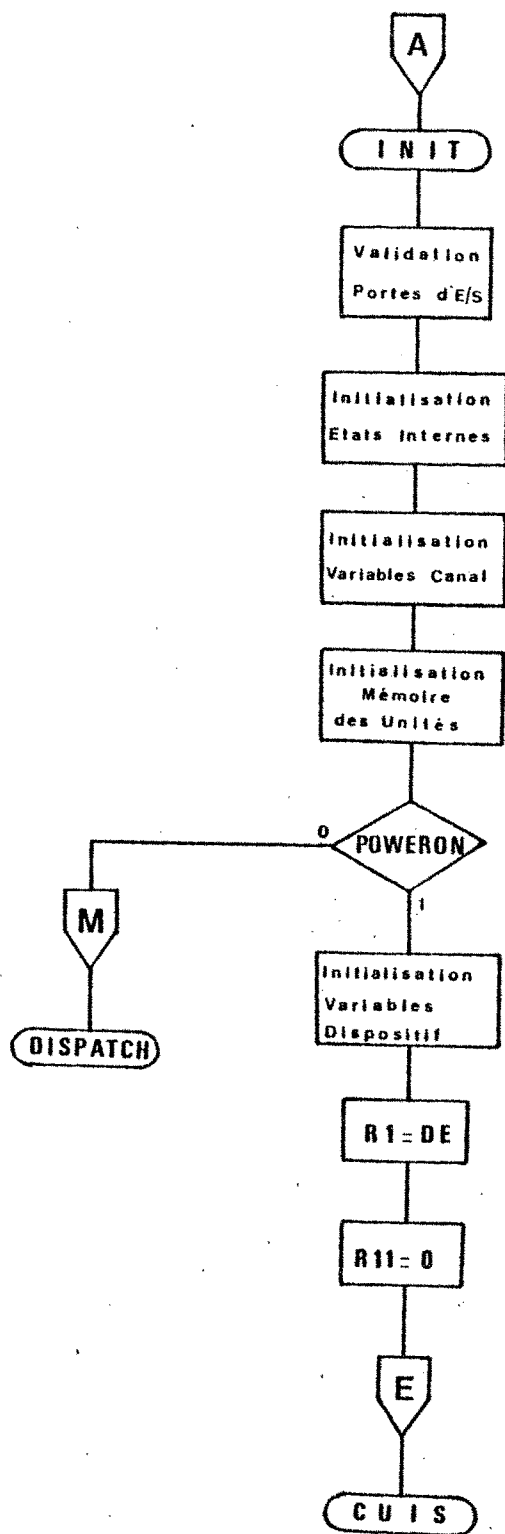


UNITE DE

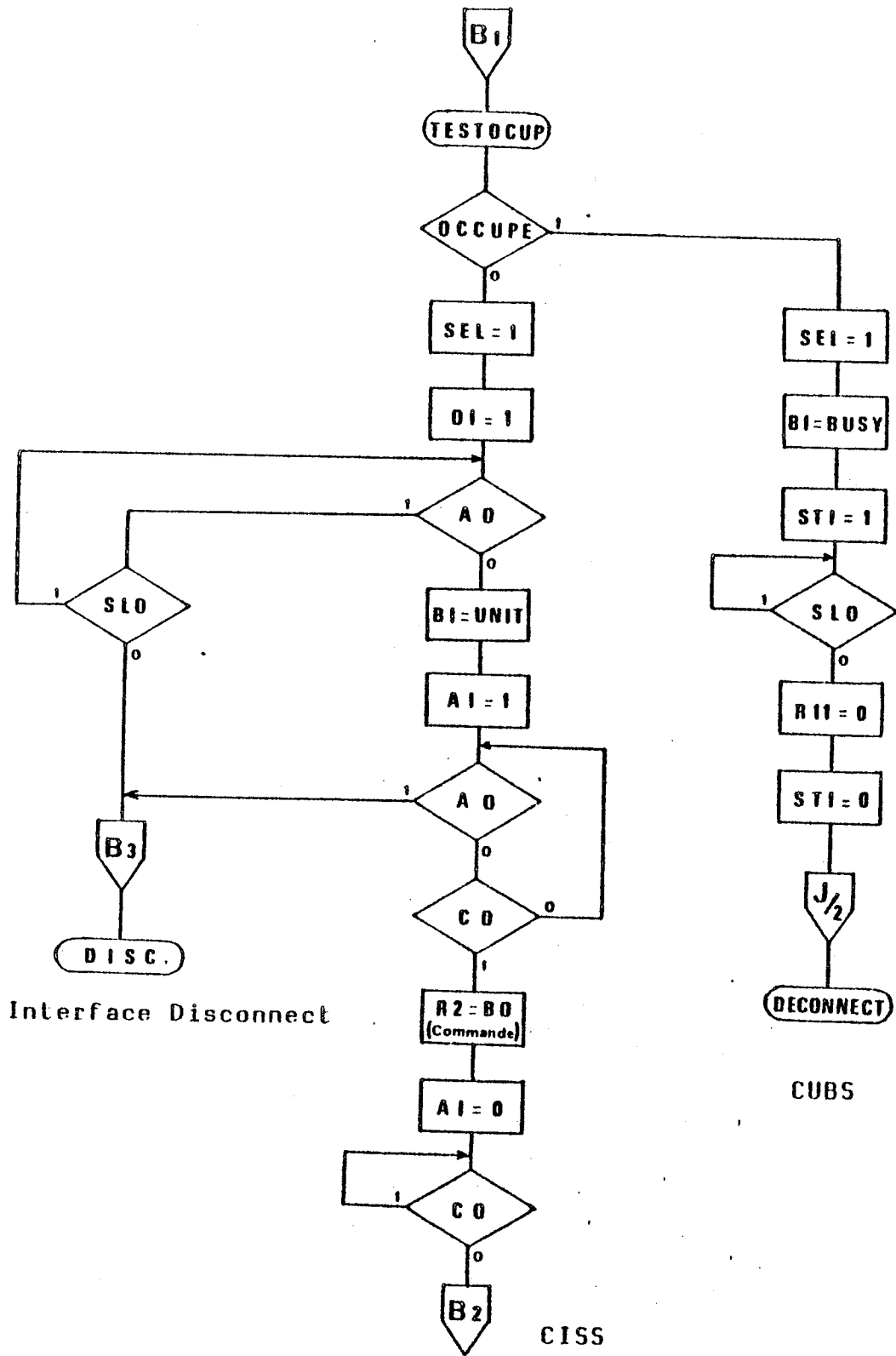
CONTROLE

BANALISE

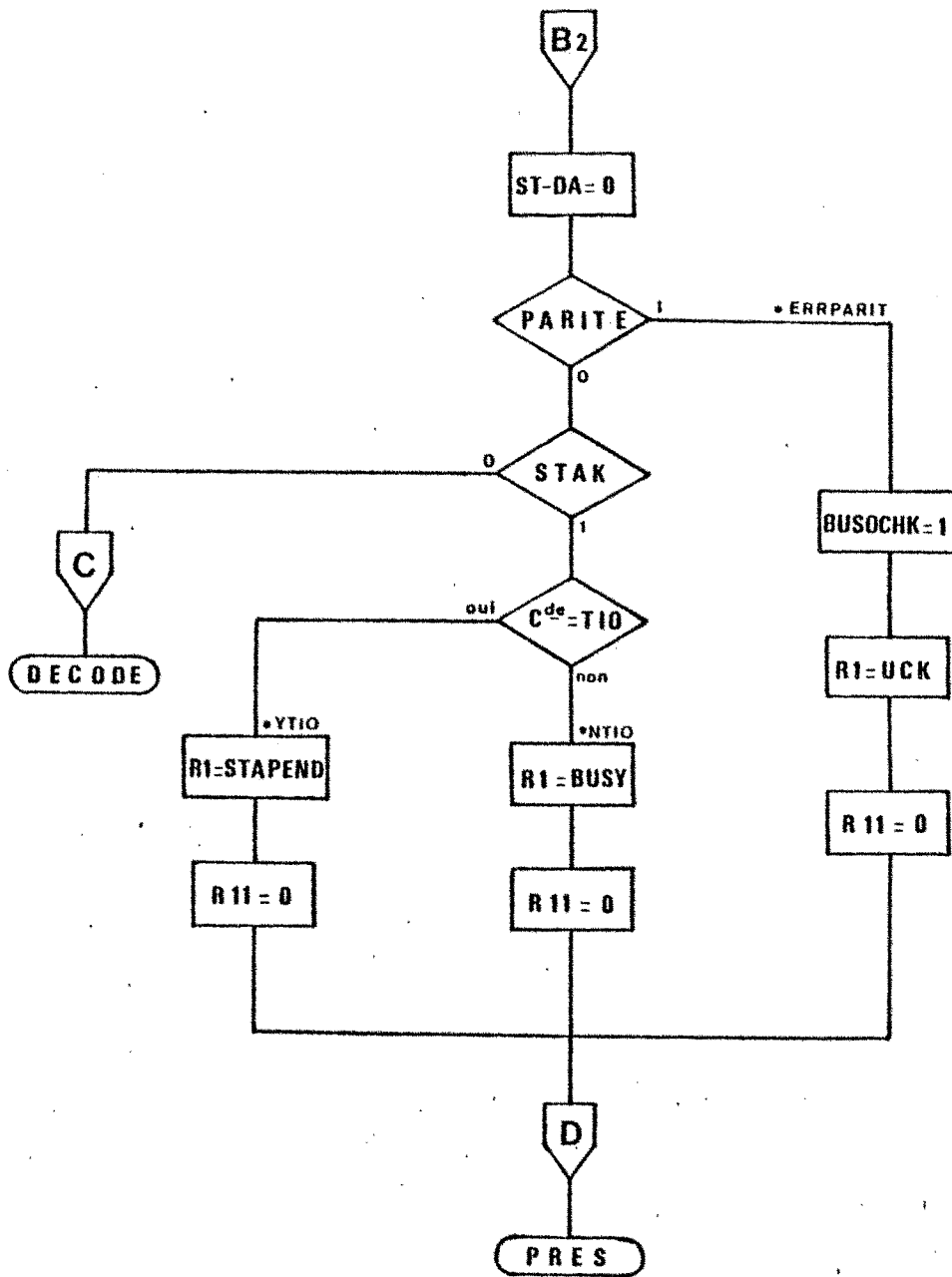




Séquence d'Initialisation



Réponses à une sélection par le Canal

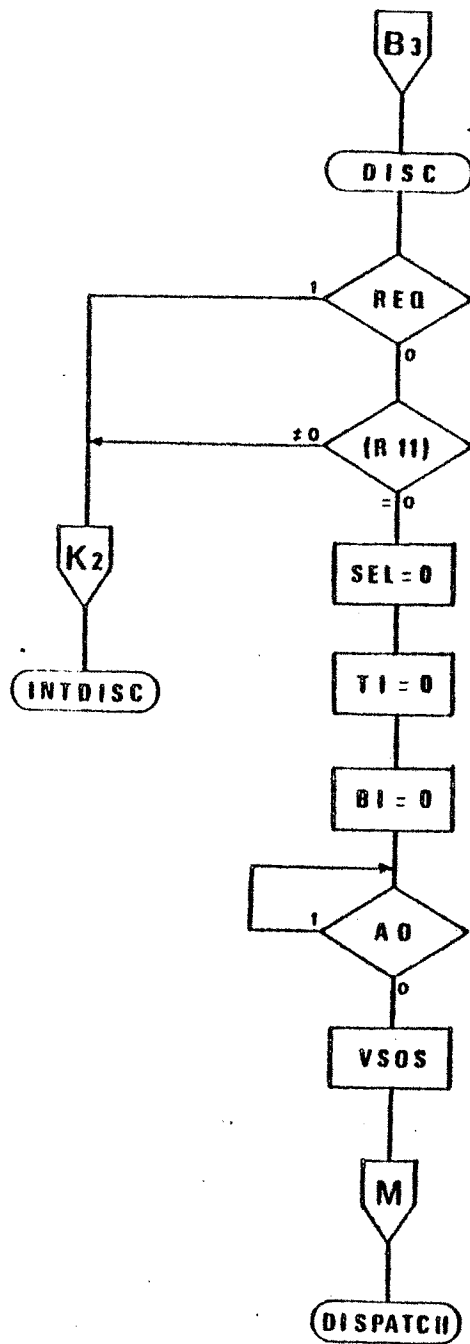


Suites d'une CISS

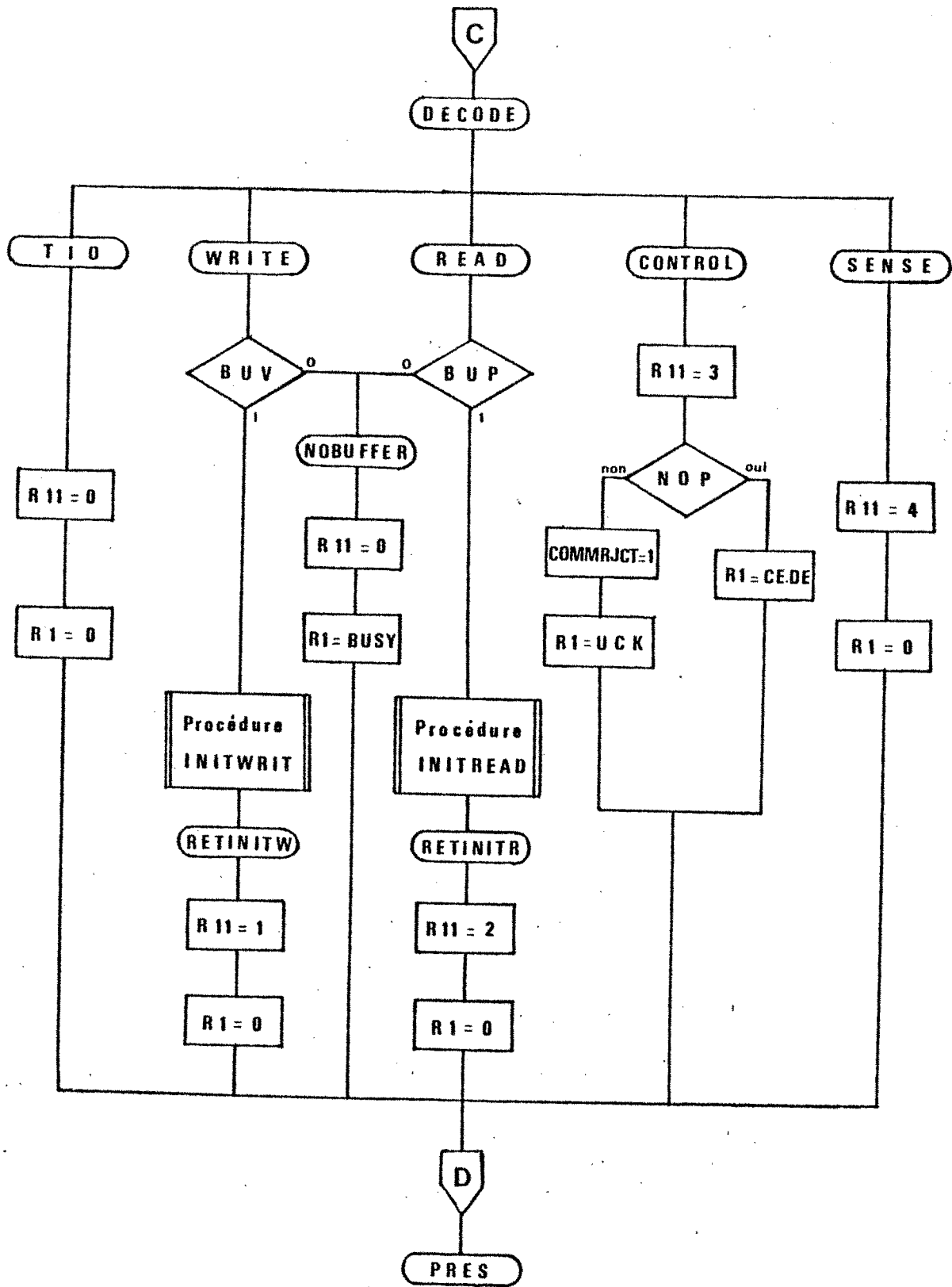
DECODE : commande acceptée

ERRPARIT : erreur de parité

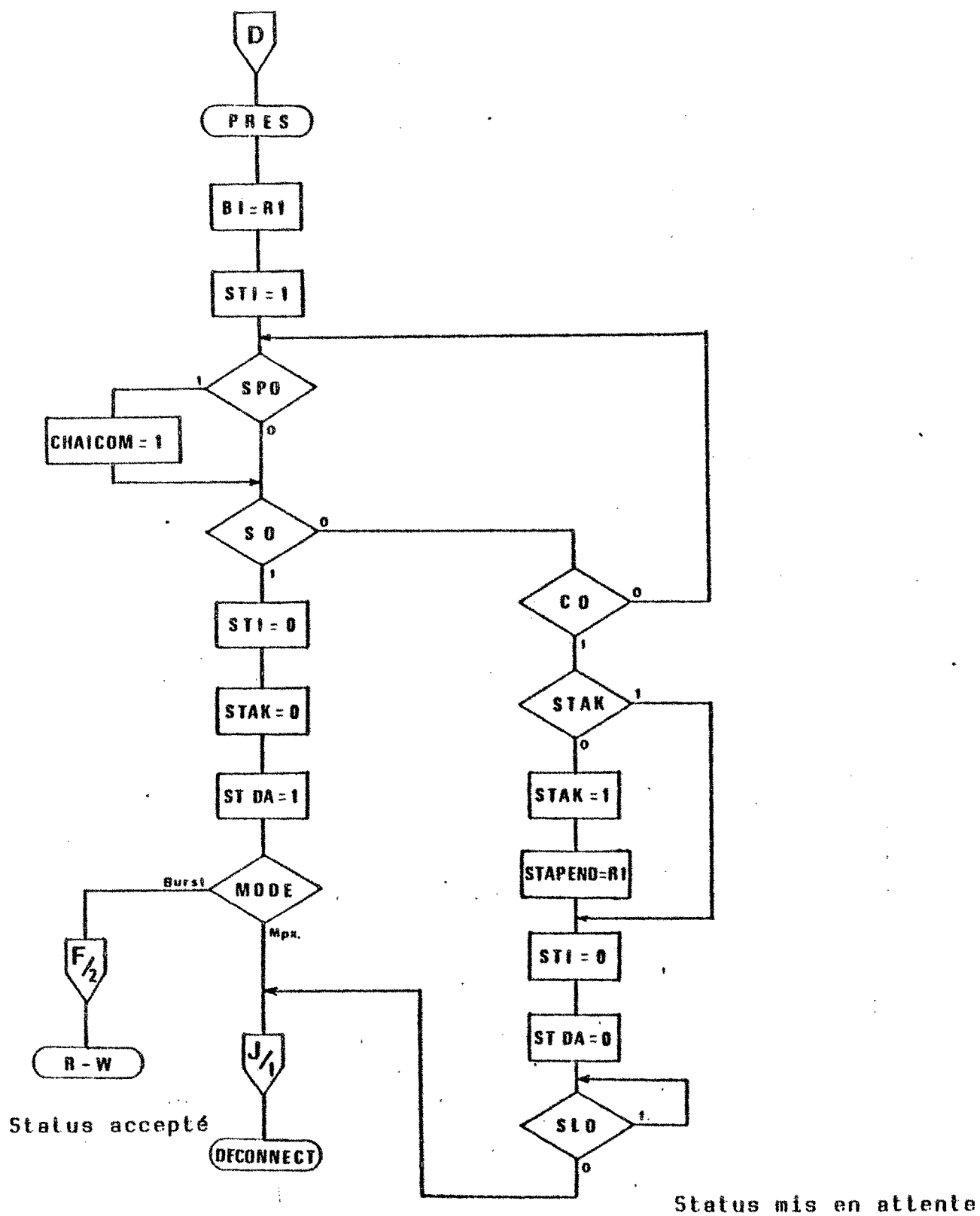
YTIO et NTIO : status en attente



Suite de Interface Disconnect

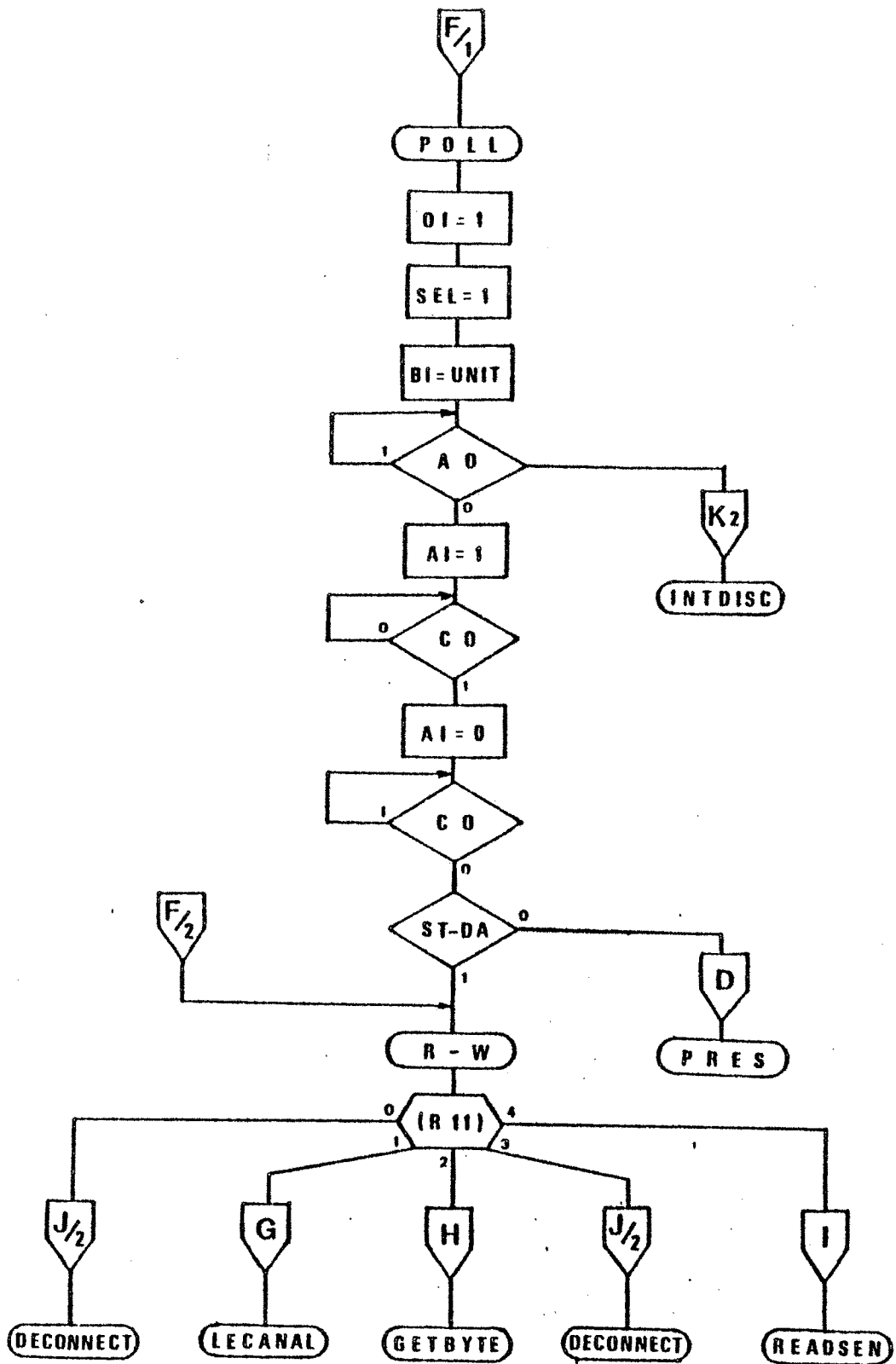


Décodage de la commande



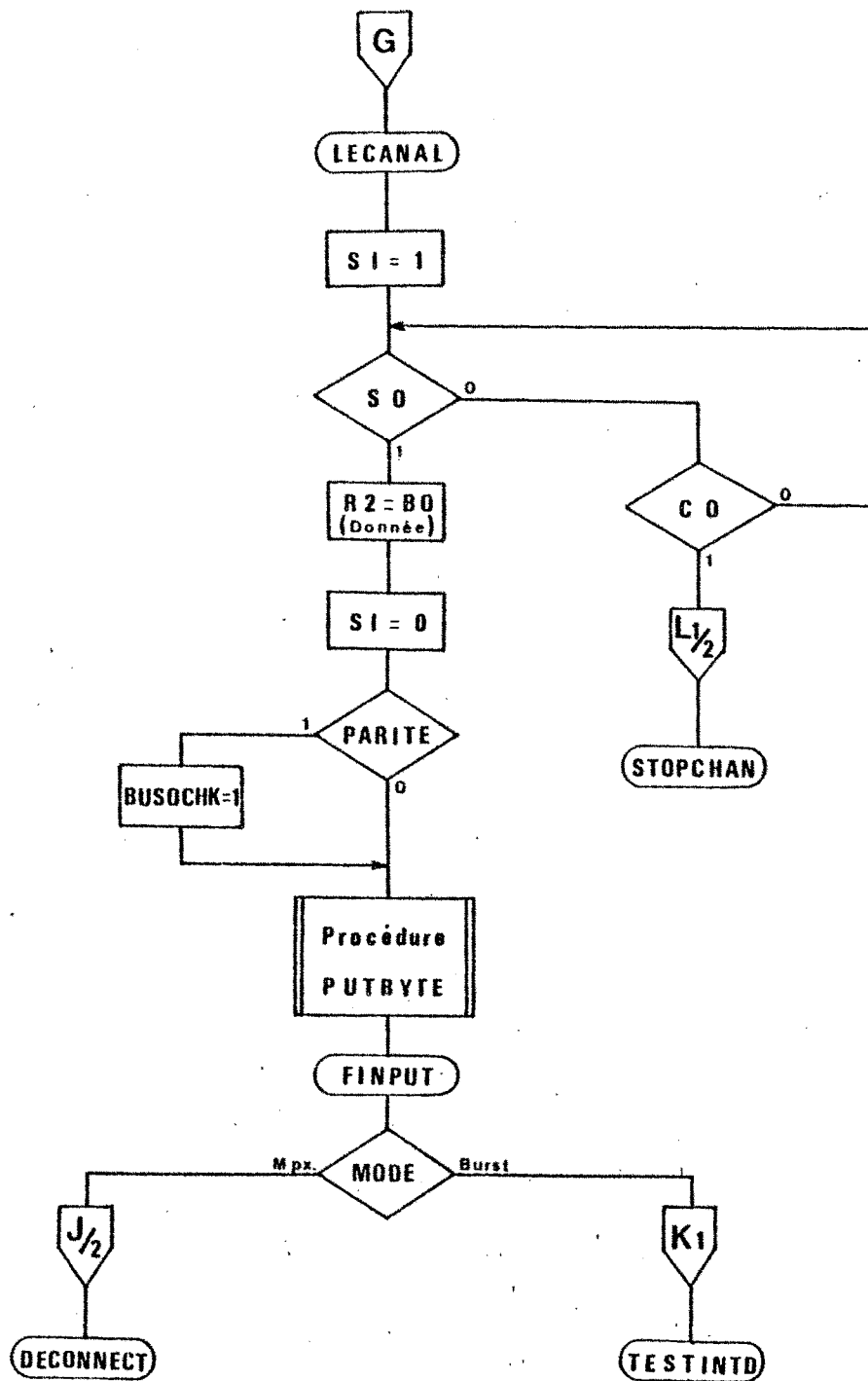
Présentation de Status



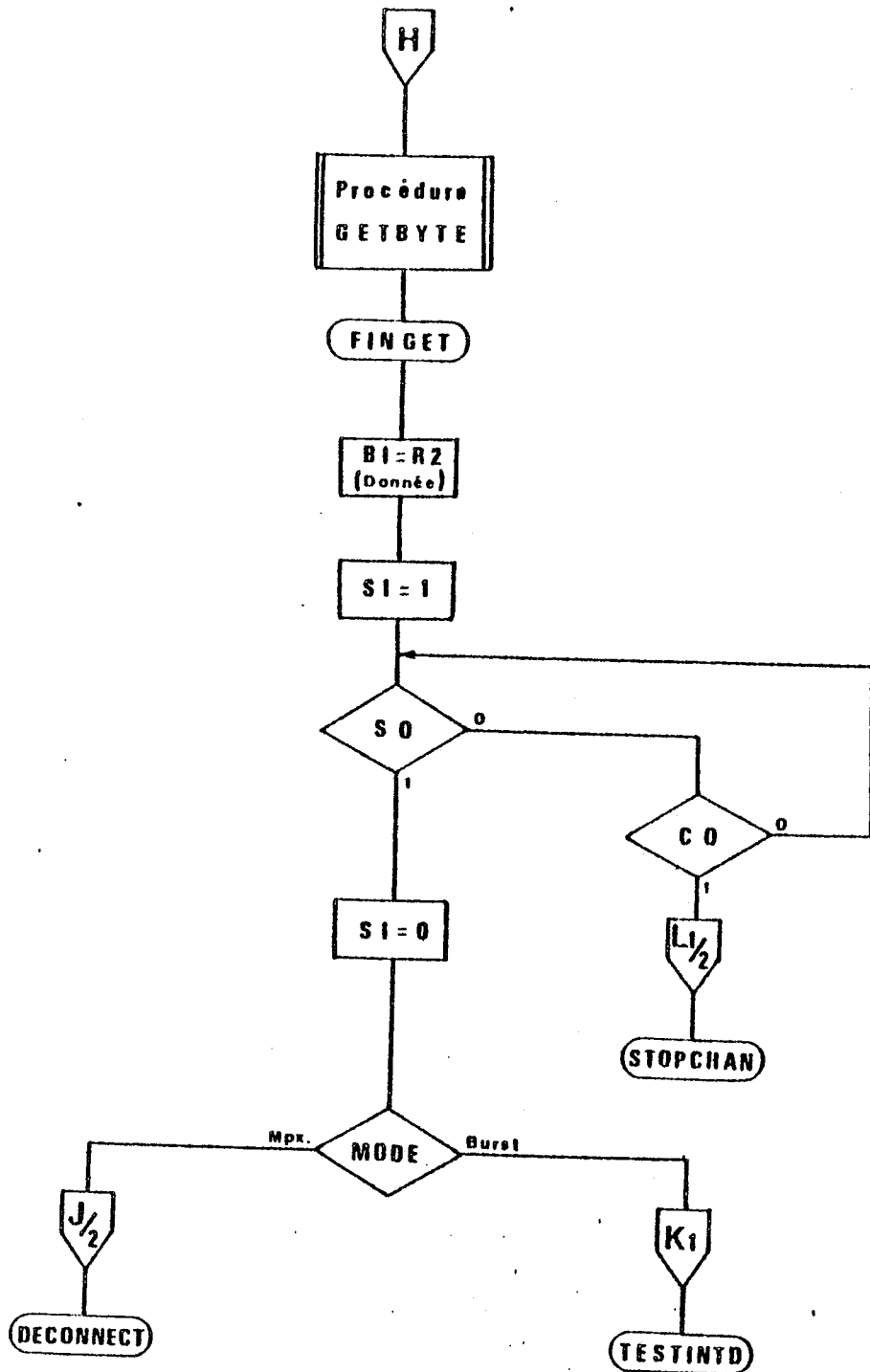


Réponse a un "polling" du Canal

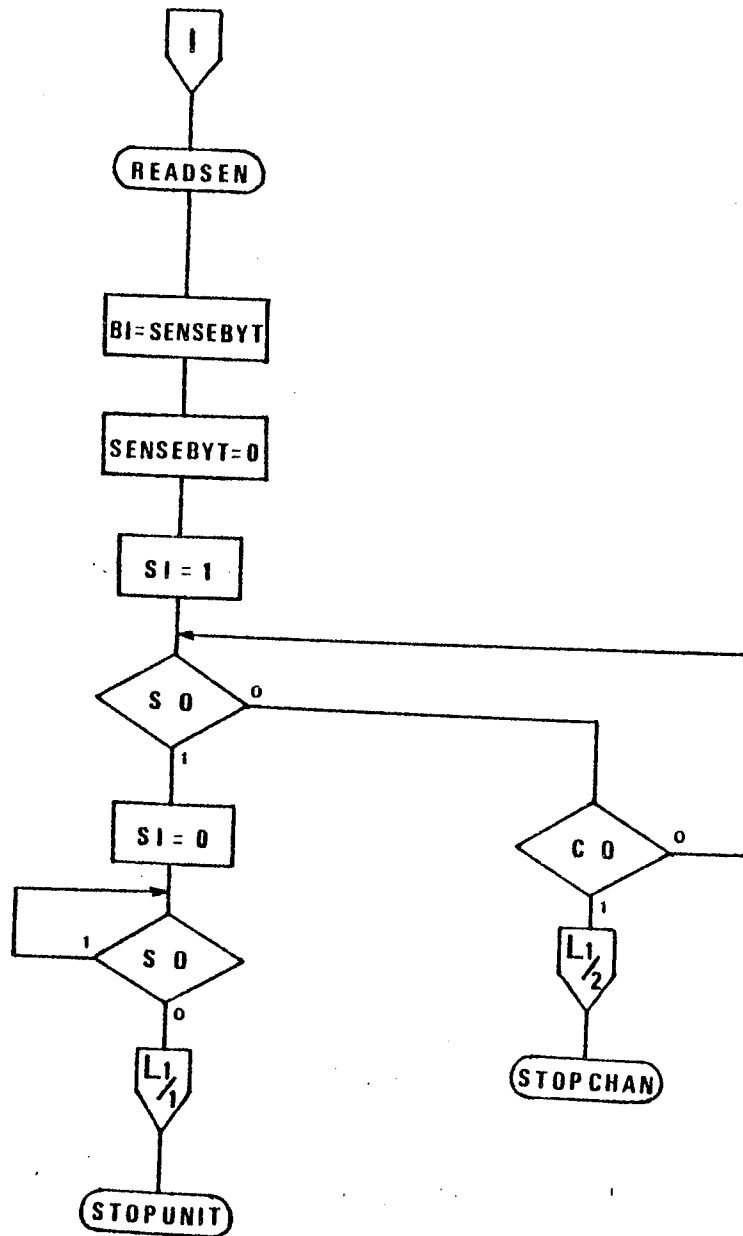




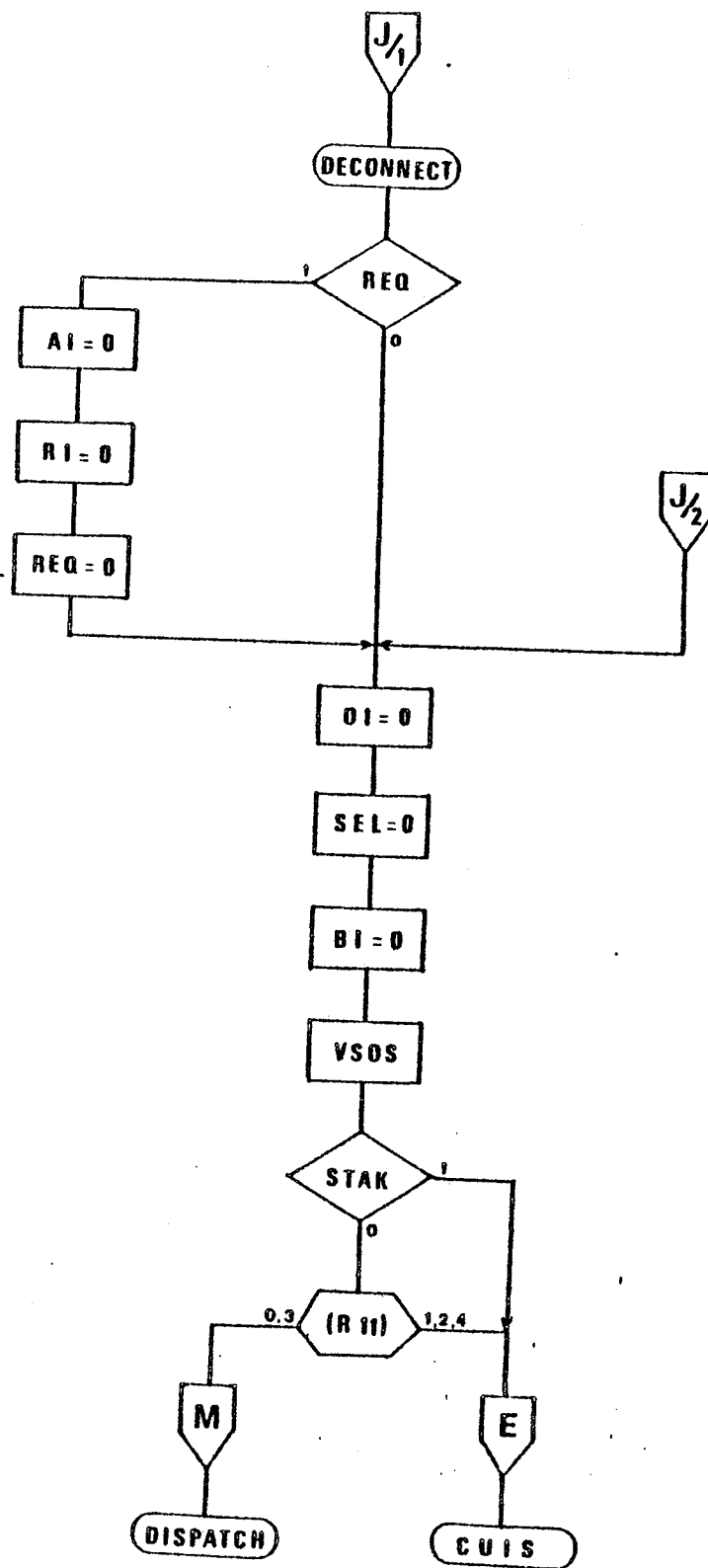
Séquence de réception d'un octet par l'UCB



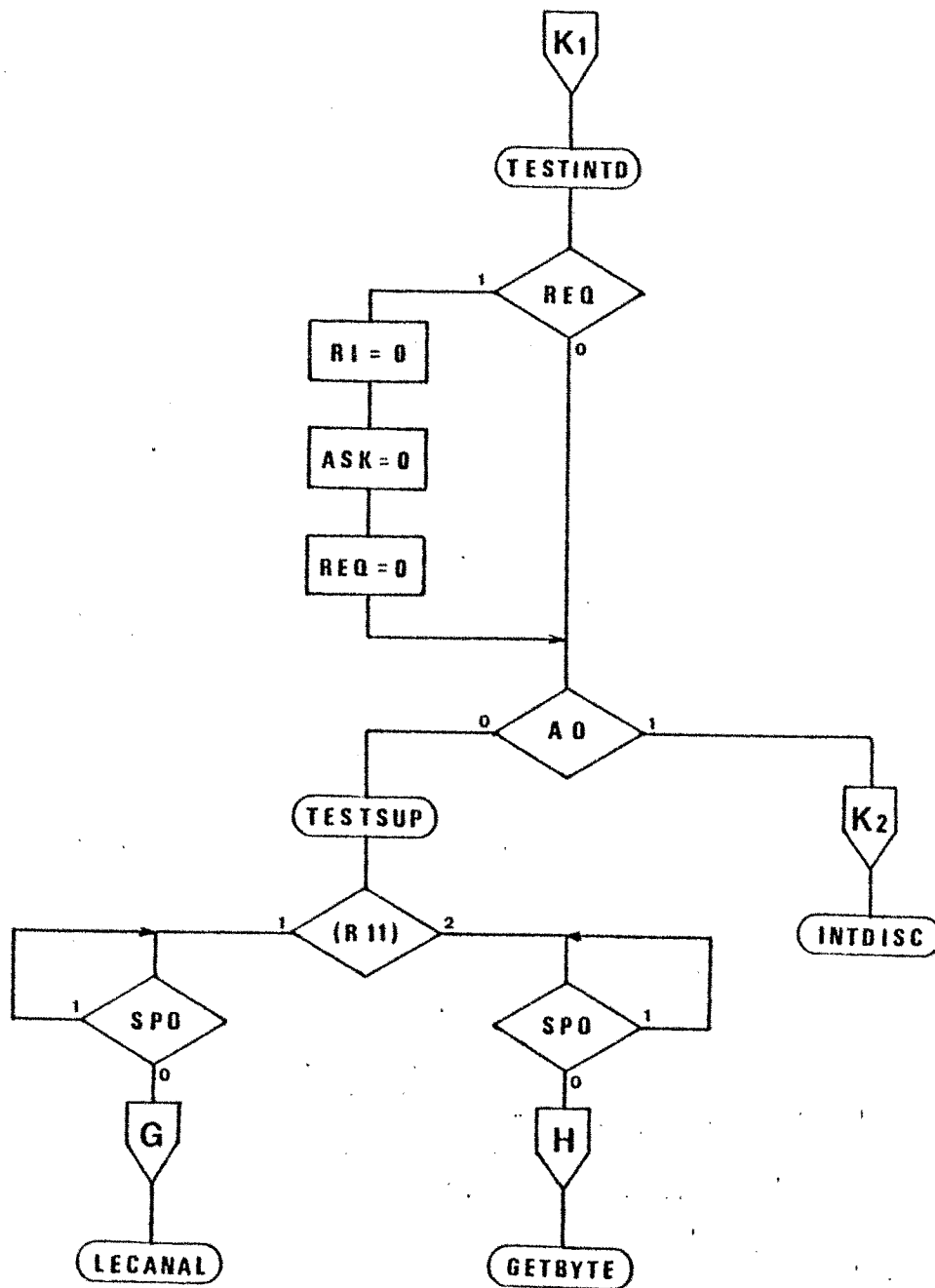
Séquence d'émission d'un octet vers le canal



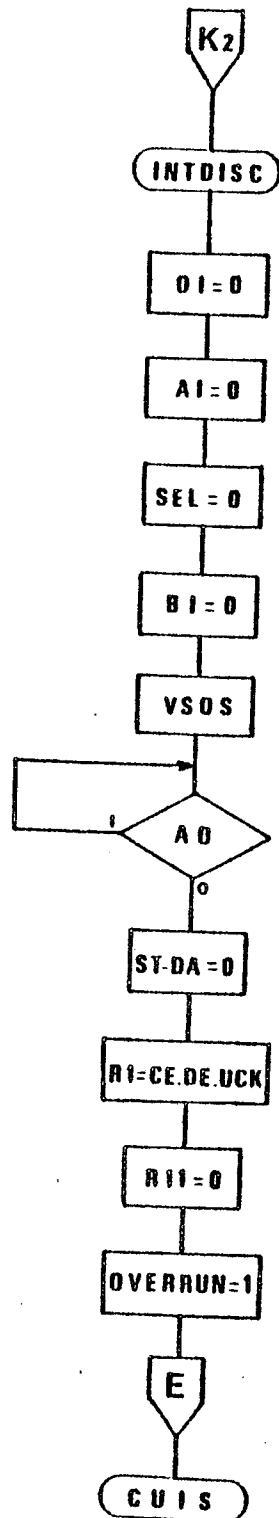
Séquence d'envoi de l'octet de SENSE vers le Canal



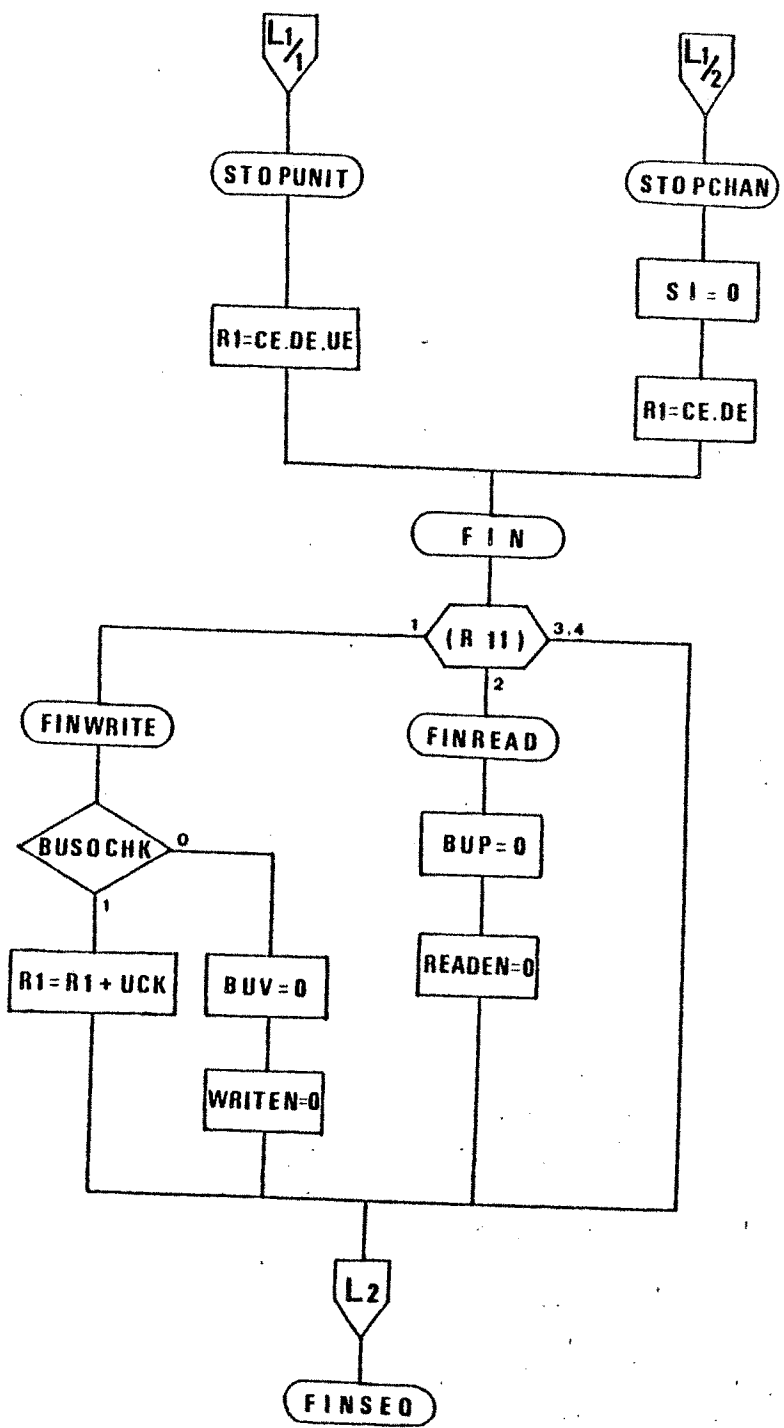
Séquence de déconnexion



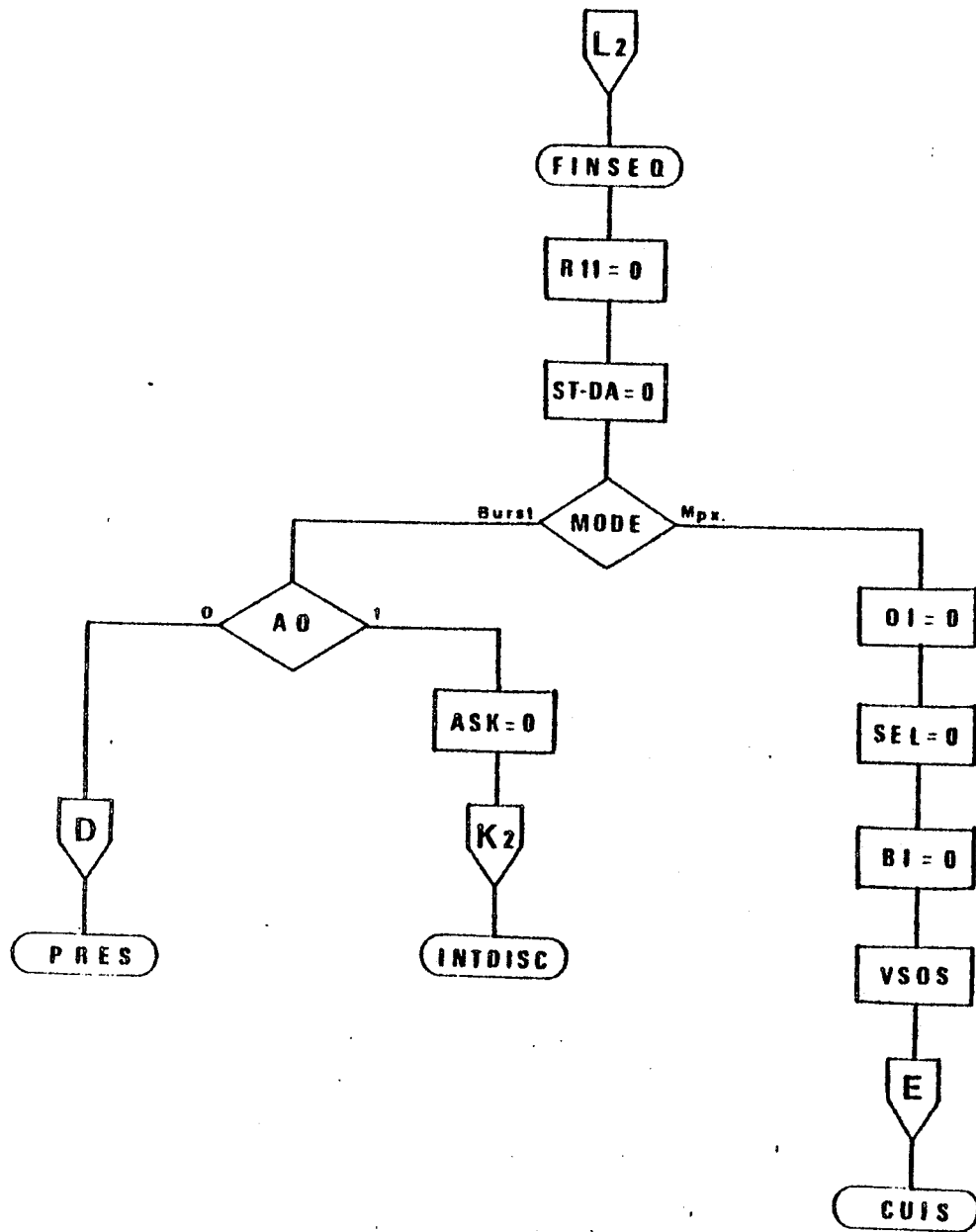
Test de "Interface Disconnect" pendant un transfert



Séquence de "Interface Disconnect" pendant un transfert

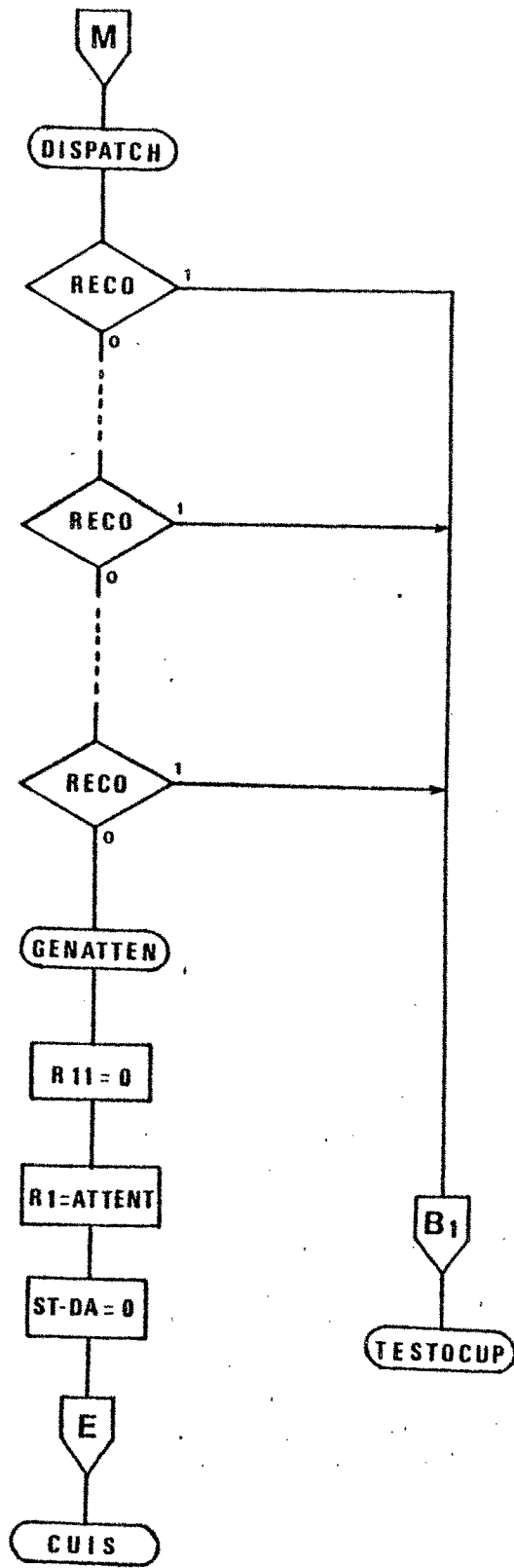


Séquence de fin de transfert

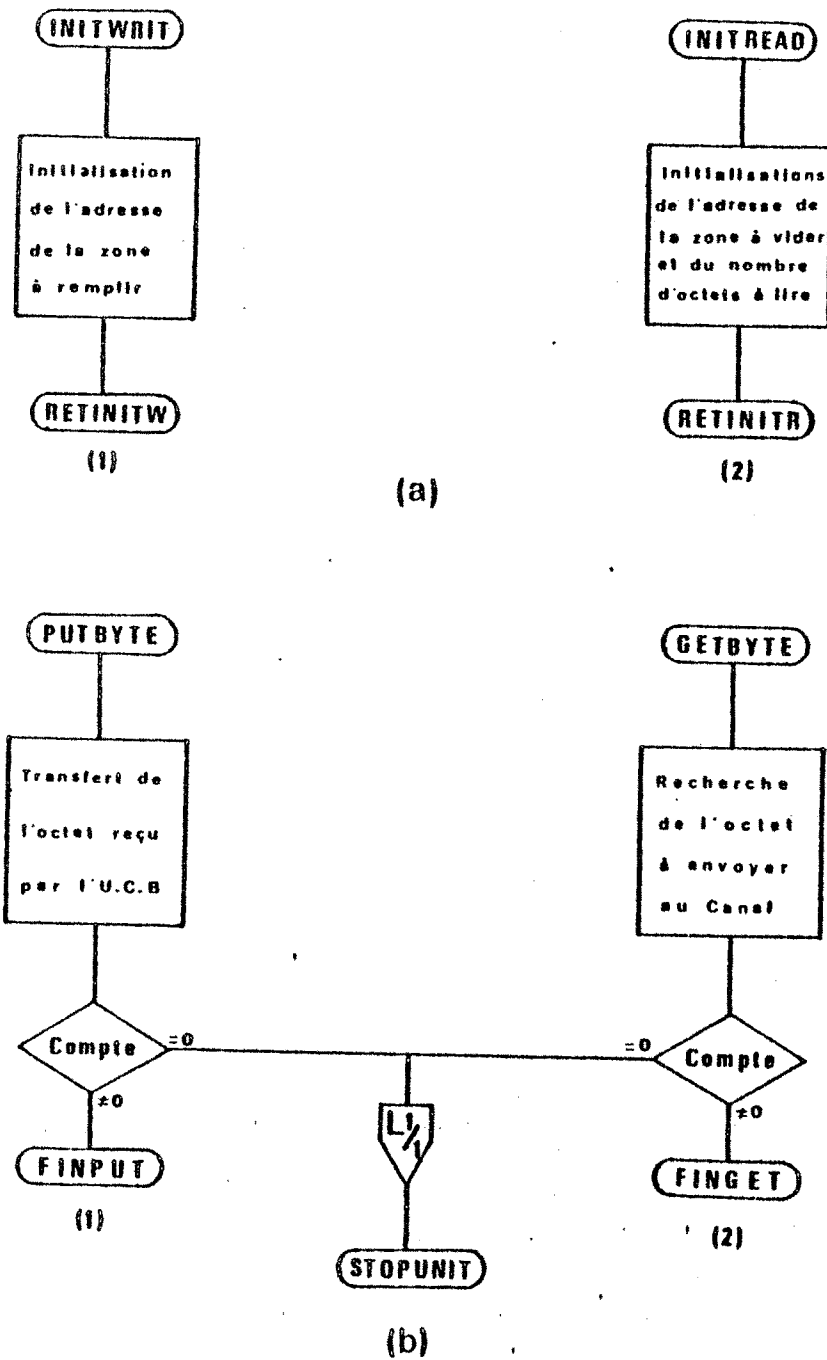


Séquence de fin de transfert suite





Structure générale du logiciel de gestion du dispositif



Sous programmes d'interface avec le dispositif



ANNEXE IVRAPPORT DE REALISATION DU SYSTEME PIAR

- 1 - Implémentation Physique
  - .1 La Maquette
  - .2 Les Modules
  
- 2 - Le Module Contrôleur de Communication
  - .1 Aspects Matériels
  - .2 Aspects Logiciels
  - .3 Experiences Réalisés
  
- 3 - Le Module Station de Transport
  - .1 Aspects Matériels
  - .2 Aspects Logiciels
  - .3 Experiences Réalisés
  
- 4 - Le Module Interface Canal
  - .1 Aspects Matériels
  - .2 Aspects Logiciels
  - .3 Experiences Réalisés
  
- 5 - La Mémoire Commune
  - .1 Aspects Matériels
  - .2 Experiences Réalisés



## 1 - IMPLEMENTATION PHYSIQUE

### 1.1 La Maquette

La maquette du système PIAR est contenue dans un châssis standard 19 pouces. Ce châssis comporte, en outre des entretoises mécaniques, les alimentations (5V,7A; +12V,0.5A; -12V,0.5A), la face avant avec les boutons, interrupteurs et différentes lampes de témoignage des alimentations et de l'activité des modules. A l'arrière du châssis se trouvent les connecteurs vers le MODEM de la ligne série et vers le bus canal IBM.

### 1.2 Les Modules

Chacun des modules du système est physiquement contenu sur une carte prototype standard No. 2 de l'Equipe de Recherche en Architecture d'Ordinateur. Le bus commun est matérialisé sur un bus banalisé de fond de panier auquel sont reliées les cartes au moyen d'un connecteur 2x49 broches.

Chaque carte contient une cinquantaine de circuits intégrés à petite, moyenne et grande échelle plus quelques composants discrets. Les connexions entre les circuits sont effectués au moyen de la technique de "wire wrapping".

## 2 - LE MODULE CONTROLEUR DE COMMUNICATION

### 2.1 Aspects Matériels

Le matériel du module contrôleur de communication MCC, est complètement réalisé et testé. Sur la carte MCC il existe 1 koctet de mémoire REEPROM contenant le superviseur SUPER-Z que nous avons écrit, 4 Koctets de mémoire vive dynamique et 128 octets de mémoire vive statique.

Le circuit SIO possède en réalité deux canaux de communication série. Nous avons utilisé un pour la liaison synchrone vers CYCLADES et l'autre pour la liaison asynchrone avec un terminal télé-imprimeur. La carte fournit vers la face avant, trois signaux qui commandent les temoins de présence de l'horloge de transmission, de la transmission des données et de la réception des données.

### 2.2 Aspects Logiciels

Le logiciel du module MCC a été complètement écrit et mis au point. Il occupe environ 1750 octets de code Z80 que sont chargées dans la mémoire dynamique à partir du terminal en utilisant les fonctions du superviseur SUPER-Z.

En plus des fonctions propres au module, nous avons inséré une fonction "mouchard" permettant d'avoir sur le terminal une trace de tous les messages (de données et de contrôle) qu'ont été reçus ou emis sur la ligne ainsi que des événements propres à la procédure (ligne "up", ligne "down", temporisations qui ont échoué, etc.).

### 2.3 Expériences Réalisés

La mise au point du logiciel a été faite en réalisant d'abord, des expériences en environnement simulé et, ensuite en environnement réel.

En environnement simulé nous avons fait les essais en bouclant la sortie de transmission des données vers la ligne sur l'entrée. Comme outils de mise au point nous avons utilisé les fonctions du superviseur SUPER-Z, l'émulateur Z80 et la sonde en temps réel d'un système de développement Tektronix 8002 et un analyseur de ligne série Biomation 110-D. Ces outils nous ont permis de mettre au point le logiciel très rapidement et de faire des mesures en temps réel.

Ces mesures ont confirmé nos calculs en ce qui concerne la valeur du débit maximal de transmission sur la ligne (96 kbits/sec) et nous ont montré que, en pratique, il était possible de le dépasser d'environ 25%. Ce résultat s'explique par les basses performances de la procédure IMM-UCIG dues à la manque d'anticipation en émission.

En environnement réel, nous avons travaillé en exploitation d'une ligne à 4800 kbits/sec. En sauvegardant les sorties du "mouchard" dans la mémoire dynamique nous avons obtenu de traces de plus de 2000 messages sans erreur de la part du MCC, ce qui nous permet d'affirmer que le logiciel est au point. En outre, nous avons provoqué différents types d'erreurs simulant les différentes erreurs qui doivent être détectées normalement par le module (temporisations d'attente, mauvais CRC, messages dupliqués, erronés ou hors de séquence).



En environnement simulé, nous avons aussi fait des essais en utilisant des procédures bi-synchrones et des procédures orientées "bit" telles que SDLC, afin de vérifier l'adaptabilité du module aux procédures plus évoluées. En utilisant la procédure SDLC nous avons exploité la ligne à des débits allant jusqu'à 250 Kbits/sec.

La figure A4-1 montre un exemple de la trace sur la ligne en environnement réel, en utilisant la procédure TMM-UCIG. Les symboles utilisés sont les suivants :

- + ligne "down"
- \* ligne "up"
- ) émission de DSM
- ( réception de ASE
- > réception de DSM
- < émission de ASE
- / émission de EOT
- % réception de EOT
- 0-7 émission bloc données (NOB = i)
- A-H réception bloc de données (NOB = i)
- : émission de ACK
- = réception de bon ACK
- I émission d'une "bulle"
- J réception d'une "bulle"
- K pas de tampons pour réception
- L "time-out" réception de ACK
- M "time-out" réception de ASE
- Q ré-émission d'un message

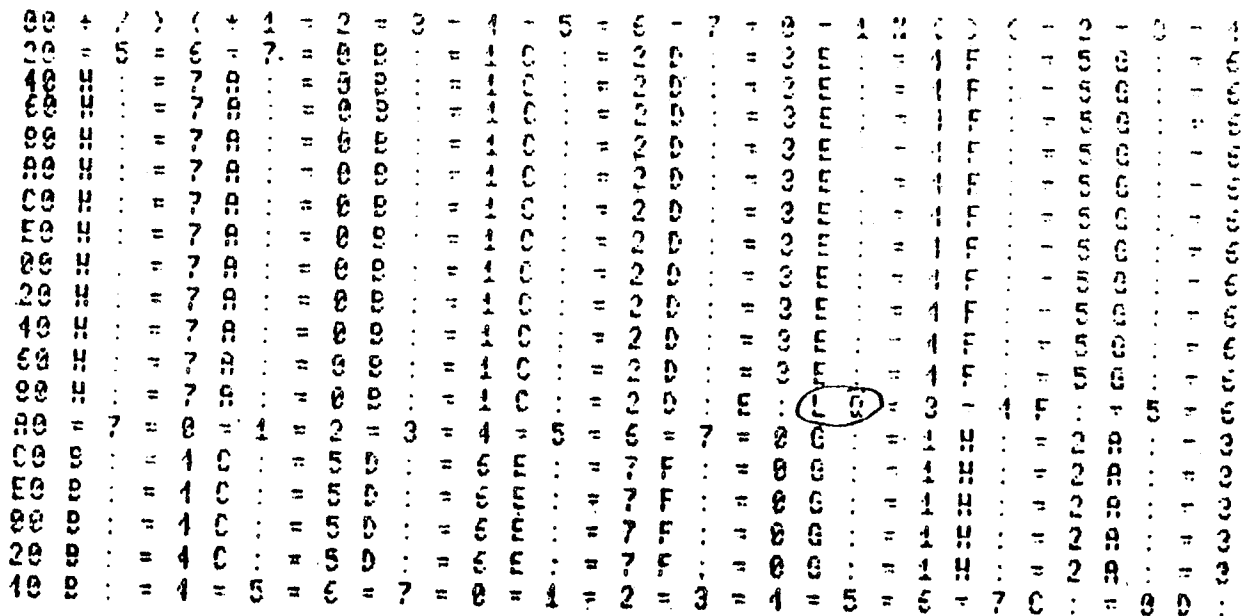


Fig. A4-1 Exemple de trace du "mouchard"

### 3 - LE MODULE STATION DE TRANSPORT

#### 3.1 Aspects Matériels

Le matériel du module station de transport MST, est complètement réalisé et testé. Sur la carte MST il existe (comme pour le module MCC), 1 koctet de mémoire morte reprogrammable contenant le superviseur SUPER-Z, 4 koctets de mémoire vive dynamique et 128 octets de mémoire vive statique. Nous avons utilisé un circuit Motorola ACIA pour faire l'interface avec le terminal télé-imprimeur.

#### 3.2 Aspects Logiciels

Le logiciel du module MST a été complètement défini, cependant il n'a pas été complètement réalisé. En effet, comme ce module n'a pas de contraintes de temps réel nous avons décidé d'écrire ce logiciel (assez important) dans un langage évolué. Cette option nous aurait permis d'écrire un logiciel d'une façon claire et nette, donc facile à mettre au point et à modifier. Cependant, le logiciel de développement correspondant pour le microprocesseur Z80 n'a pas été disponible dans les délais prévus. D'autre part, le logiciel résiduel sur l'ordinateur hôte n'a pas été écrit (ce qui n'était pas dans le cadre de notre projet).

Actuellement nous sommes en train d'écrire une version réduite, mono-utilisateur, du protocole de transport et un noyau du logiciel correspondant sur l'IBM 360/67.

#### 3.3 Expériences Réalisées

La mise au point du module Station de Transport nous a permis de tester extensivement la communication entre les modules au moyen de la mémoire commune et l'algorithme d'allocation des tampons. Ces tests nous les avons faits à l'aide des fonctions

fournies par le superviseur SUPER-Z.

Après avoir vérifié que la communication entre modules se déroulait comme prévu, nous avons commencé à faire des essais de communication en environnement réel avec les stations de transport distantes. Pour cela nous avons utilisé les adresses des stations de transport "écho" afin de vérifier le comportement de la SI Externe vis-à-vis des SI distantes. Pour ces essais nous avons utilisé aussi les fonctions fournies par le superviseur SUPER-Z. Dans ces conditions nous avons ouvert des flots, avec et sans services additionnels, et nous avons bien reçu en écho les paquets de données que nous avons envoyés.

## 4 - LE MODULE INTERFACE CANAL

### 4.1 Aspects Matériels

Le matériel du module d'interface avec le canal IBM est complètement réalisé et testé. Sur la carte il existe 512x32 bits de mémoire morte contenant les programmes du module. Cependant, nous avons laissé les emplacements et connecteurs afin de pouvoir utiliser l'outil de mise au point MADAM. La carte fournit, vers la face avant trois signaux qui indiquent l'état du module vis-à-vis du canal: PRET, DEMANDEUR et SELECTE. Un commutateur permet de sélectionner le mode d'opération de l'unité : MULTIPLEX, BURST ou MIXTE.

### 4.2 Aspects Logiciels

Le logiciel du module MIC est complètement réalisé et mis au point. La modularité du logiciel nous a permis de développer et mettre au point séparément le logiciel d'interface avec le canal et d'interface avec la mémoire de communication avec les autres modules du système. Ensuite nous avons développé le moniteur qui gère la communication entre les deux parties. Pendant la mise au point, les programmes étaient contenus dans la mémoire vive de l'outil MADAM, ce qui nous a permis de les charger, modifier, placer des points d'arrêt, etc.. Actuellement, les programmes ont été inscrits sur des mémoires mortes programmables PROM.

### 4.3 Expériences Réalisées

La mise au point du module Interface Canal s'est avérée assez difficile car il a fallu faire les essais et la mise au point en dehors des heures normales d'exploitation de l'ordinateur IBM 360/67.

Dans un premier temps nous avons fait des essais sur le canal IBM avec une machine "nue", c.a.d. sans un système d'exploitation

mais avec un noyau moniteur qui dialogue avec la console opérateur et accepte un nombre limité de commandes. Ces commandes nous permettent de lancer des opérations élémentaires d'entrée/sortie.

Pour la mise au point du matériel et des programmes nous avons utilisé extensivement deux outils : un analyseur digital Tektronix 7D01 et l'outil de développement MADAM. Ces outils nous ont permis d'observer toutes les séquences de dialogue avec le canal et en particulier un certain nombre qui n'étaient pas bien documentés dans la notice du constructeur.

Une fois que nous avons constaté que le dialogue avec le canal se déroulait normalement, nous avons fait des essais sous le système d'exploitation CP/67, à partir d'une machine virtuelle sous CMS. A ce moment la, nous avons rencontré un certain nombre de difficultés dues au fait que c'est le système d'exploitation qui fait la gestion des entrées/sorties et elles ne sont pas toutes réfléchies vers la machine virtuelle. Nous avons aussi constaté qu'en cas d'erreur, CP/67 génère des commandes et séquences de sa propre initiative. La mise au point se montra alors plus difficile car il était plus lent de récupérer les "cassures de système" qui occasionnaient les erreurs éventuelles.

Actuellement, le module peut tourner en exploitation normal du système sans occasionner des pannes ni des défaillances (y compris à la mise sous-tension et à la mise hors-tension).

```

.tio
UNIT= 12
NOTOPER
..sense
IT 0012 00000000 04000000
IT 0012 00000000 80000000
UNIT= 12
STARTED
IT 0012 00000560 00000000
01F8F7F6 03000000 F1F2C1C2
.tio
UNIT= 12
UNITLIBR
.read
COUNT=0A
UNIT= 12
STARTED
IT 0012 00000550 0C400000
IT 0012 00000000 80000000
F9F8F7F6 03000000 F1F2C1C2
.write
COUNT=10
UNIT= 12
STARTED
IT 0012 00000558 0C400000
F9F8F7F6 03000000 F1F2C1C2

```

Exemple de sortie sur la console opérateur sous machine "nue"

```

. tio
LIBRE
. sense
OCTET DE SENSE = 00
STATUS RANGE = 000C
. read
COMPTE ?
10
STATUS RANGE
STATUS UNITE = 02. STATUS CANAL = 00.
. read
COMPTE ?
10
STATUS RANGE
STATUS UNITE = 02. STATUS CANAL = 00.
. read
COMPTE ?
50
STATUS RANGE
STATUS UNITE = 02. STATUS CANAL = 00.
. sense
OCTET DE SENSE = 40
STATUS RANGE = 000C

```

Exemple de sortie sur un terminal sous CP/CMS

## 5 - LA MEMOIRE COMMUNE

### 5.1 Aspects Matériels

Le matériel de la mémoire commune de communication entre les modules, ainsi que celui de l'allocateur cablée du bus partagé sont complètement réalisés et au point. La mémoire est constitué par 4 Koctets de mémoire vive statique.

### 5.2 Experiences Réalisés

Nous avons testé deux versions de l'allocateur de bus : une synchrone et l'autre asynchrone. Pour les mesures nous avons utilisé un oscilloscope rapide (100 MHz) dans les cas repetitifs et l'analyseur diigital Tektronix 7D01 pour les séquences non repetitives asynchrones.

Dans la version synchrone, les trois microprocesseurs du système utilisaient un même horloge de 8 MHz fournit par le 8X300 et ensuite divisé à 2 MHz pour les deux microprocesseurs Z80. Cet horloge est aussi utilisé pour l'échantillonnage des bascules de l'allocateur. Il est a noter que un accès a la mémoire commune par le microprocesseur 8X300 est constitué par l'exécution de plusieurs instructions.

Dans ces conditions nous avons fait des essais extensifs en creant des conflits par des accès simultanés repetitifs des trois microprocesseurs. Nous avons confirmé la priorité des modules au moment des conflits et que le module de priorité plus faible (MST) se voit allouer le bus dans un temps finit, inferieur à 1.5 microsecondes, même dans de cas ou les autres processeurs font des accès repetitifs.

Dans la version asynchrone, chaque microprocesseur utilisait un horloge propre à lui, donc mutuellement asynchrones. Nous avons alors généré l'horloge d'échantillonnage à partir des



transitions des signaux des requêtes des modules. Nous avons fait les mêmes essais que pour la version synchrone, avec des résultats similaires. Nous n'avons pas pu constater un cas de meta-stabilité du circuit de mémorisation de l'allocateur, cependant nous n'avons pas fait des essais extensifs dans ce sens.

AUTORISATION DE SOUTENANCE

VU les dispositions de l'article 3 de l'arrêté du 16 Avril 1974,

VU les rapports de présentation de Messieurs :

- F. ANCEAU, Maître de Conférences à l'Institut National Polytechnique de GRENOBLE
- H. ZIMMERMANN, Responsable de Recherche en Informatique Répartie - I.R.I.A. LE CHESNAY -

Monsieur Roberto Jaime SAETTONE SUMMERS

est autorisé à présenter une thèse en soutenance pour l'obtention du diplôme de DOCTEUR-INGENIEUR, spécialité "Génie Informatique".

Grenoble, le 29 Juin 1979

Le Président de l'I.N.P.G.

**Ph. TRAYNARD**  
Président  
de l'Institut National Polytechnique