

50376  
1988  
5

50376  
1988  
5

N° d'ordre 1391

THESE

présentée à

L'UNIVERSITE DES SCIENCES ET TECHNIQUES DE LILLE

pour l'obtention du titre de

DOCTEUR DE 3ème CYCLE

Spécialité: Automatique

par

DOMINE FRANÇOIS



**INTEGRATION D'UN MONITEUR CARDIORESPIRATOIRE SUR UNE  
ARCHITECTURE MULTIPROCESSEURS**

Soutenue le 12 février 1988 devant la Commission d'Examen

Mr VIDAL	Président et Rapporteur
Mr TOULOTTE	Examineur
Mr POSTAIRE	Examineur
Mr STAROSWIECKI	Examineur
Mr MOSCHETTO	Examineur
Mme CHAMBRIN	Examineur

SCD LILLE 1



D 030 318081 5

50376  
1988  
5

50376  
1988  
5

AVANT PROPOS



Le travail présenté dans ce mémoire a été effectué dans le cadre de l'INSERM de Lille et du Centre d'Automatique des Sciences et Techniques de Lille Flandres Artois.

Il est l'aboutissement de la collaboration, des aides, des conseils et des encouragements qui m'ont été apportés par d'éminents spécialistes sans qui il n'aurait pas pu être réalisé. Je tiens à leur exprimer ma plus vive gratitude, et à leur présenter mes remerciements:

à Monsieur le Professeur P. VIDAL, Directeur du Centre d'Automatique, pour l'honneur qu'il me fait en présidant ce jury et en acceptant d'être le rapporteur de l'ensemble de ces travaux. Ses encouragements et les multiples conseils qu'il m'a prodigués ont été un puissant soutien dans mes diverses activités,

à Monsieur le Professeur J.M. TOULOTTE du centre d'Automatique, qui a bien voulu faire une prélecture objective de ces travaux,

à Monsieur le Professeur J.G. POSTAIRE, du centre d'Automatique, dont l'aide technique a favorisé l'aboutissement de ce travail,

à Monsieur le Professeur M. STAROSWIECKI, du centre d'Automatique, qui a eu l'amabilité de juger la qualité de ce travail avec bienveillance,

à Monsieur le Professeur Y. MOSCHETTO, Directeur de recherches INSERM, pour l'intérêt qu'il a témoigné à ces travaux,

à Madame M.C. CHAMBRIN, chargée de recherche à l'INSERM qui a suivi cette étude dans son ensemble, et m'a apporté le soutien de sa compétence pour amener ce travail à son niveau actuel,

Enfin, à tous mes collègues et amis qui m'ont soutenu, ainsi qu'au personnel du laboratoire, qui a participé avec gentillesse à la confection de ce mémoire.

## INTRODUCTION GENERALE

### I) Présentation du projet

#### I-1) Cahier des charges

- I-1-1) Signaux du bilan
- I-1-2) Modularité
- I-1-3) Fonctions de l'appareil

#### I-2) Méthode de Calvez

- I-2-1) Définition des spécifications
- I-2-2) Structure fonctionnelle
- I-2-3) Description opératoire
- I-2-4) Structure exécutive
- I-2-5) Réalisation

- I-2-5-1) Réalisation matérielle
- I-2-5-2) Réalisation logicielle

#### I-3) Traitement du signal

##### I-3-1) Introduction

##### I-3-2) Analyse du signal

- I-3-2-1) Mémorisation et traitement
- I-3-2-2) Contraintes temps réel

##### I-3-3) Conclusion

#### I-4) Application au traitement de l'E.C.G.

- I-4-1) Introduction
- I-4-2) Description de l'électrocardiogramme
- I-4-3) Troubles du rythme
- I-4-4) Traitement
- I-4-5) Conclusion

## I-5) Signaux respiratoires

- I-5-1) Introduction
- I-5-2) Présentation des signaux
- I-5-3) Calcul des paramètres
- I-5-4) Etude des paramètres
  - I-5-4-1) Signal de pression
  - I-5-4-2) Signal de volume
  - I-5-4-3) Signal de CO<sub>2</sub>
  - I-5-4-4) Paramètres calculés

## I-6) Exploration des pressions cardio-vasculaires

- I-6-1) Rappels physiologiques
- I-6-2) Capteurs
- I-6-3) Morphologie
- I-6-4) But du traitement

## I-7) Conclusion

## II) Aspects fonctionnels

### II-1) Structure micro-informatisée

- II-1-1) Introduction
- II-1-2) Description d'un bus
- II-1-3) Types de liaisons
- II-1-4) Choix du matériel
- II-1-5) Fonctionnement du Multibus
- II-1-6) Commande décentralisée du Multibus
- II-1-7) Conclusion

### II-2) Mémoire de masse

### II-3) Système d'exploitation

- II-3-1) Introduction
- II-3-2) Modèle d'un système d'exploitation



## II-3-3) Principes de base

- II-3-3-1) Introduction
- II-3-3-2) Noyau
- II-3-3-3) Processus
- II-3-3-4) Gestion des ressources
- II-3-3-5) Exclusion mutuelle
- II-3-3-6) Sémaphores
- II-3-3-7) Communication entre processus
- II-3-3-8) Gestions des échanges
- II-3-3-9) Gestion de la mémoire
- II-3-3-10) Langage de commande
- II-3-3-11) Conclusion



## II-4) Système de développement

## II-5) Phases de développement

## II-6) Conclusion

## III) Dialogue homme-machine

### III-1) Introduction

### III-2) Présentation des résultats

- III-2-1) Imagerie
- III-2-2) Variables visuelles
- III-2-3) Histogramme

### III-3) Conclusion

### III-4) Dialogue Homme-Machine

- III-4-1) Introduction
- III-4-2) Interactivité
- III-4-3) Contraintes logicielles
- III-4-4) Conclusion

#### IV) Réalisation pratique

- IV-1) Introduction
- IV-2) Synoptique général
- IV-3) Carte C.A.D.
- IV-4) Carte B.A.L.
- IV-5) Modules Mm
- IV-6) Modules Mi
- IV-7) Modules Ms



- IV-7-1) Description
- IV-7-2) Rôle
- IV-7-3) Fonctionnement global

- IV-8) Console de dialogue
- IV-9) Conclusion

#### V) Fonctionnement de l'appareil

##### V-1) Console de pilotage

- V-1-1) Introduction
- V-1-2) Représentation des résultats
- V-1-3) Fonctionnement
- V-1-4) Présentation de la page 1
- V-1-5) Présentation de la page 2
- V-1-6) Présentation de la page 3
- V-1-7) Description des commandes

- V-1-7-1) INIT
- V-1-7-2) MENU
- V-1-7-3) PREC et SUIV
- V-1-7-4) TEXT
- V-1-7-5) SIGN
- V-1-7-6) ALAR
- V-1-7-7) EXAM

V-1-8) Présentation de la page 4

V-1-9) Conclusion

## V-2) Présentation générale du logiciel

- V-2-1) Conception modulaire
- V-2-2) Archivage et accès aux données
- V-2-3) Partage du processeur



- V-2-3-1) Exécution des programmes
- V-2-3-2) Modèle de file d'attente
- V-2-3-3) Synchronisation temporelle

## V-2-4) Description des échanges

- V-2-4-1) Communication Ms -> Console
- V-2-4-2) Communication Console -> Ms
- V-2-4-3) Communication C.A.D. -> Mi
- V-2-4-4) Communication Mi -> Ms

## V-3) Conclusion

## VI) Perspectives

### VI-1) Modularité

- VI-1-1) Interconnexion nouvelles cartes
- VI-1-2) Problèmes logiciels
- VI-1-3) Problèmes matériels

### VI-2) Evolution future

- VI-2-1) Améliorations matérielles
- VI-2-2) Conclusion

### VI-3) Proposition pour l'implantation dans un service de traumatologie d'urgence

### VI-4) Conclusion

## CONCLUSION GENERALE

INTRODUCTION GENERALE

L'objectif de ce travail est la réalisation d'un système informatisé individuel de surveillance des malades en réanimation.

Actuellement, l'équipement d'une chambre de réanimation est caractérisé par l'importance et la diversité des appareils nécessaires à la surveillance des signaux physiologiques d'origine cardio-vasculaire et respiratoire. Les appareils de monitoring possèdent des champs d'action limités à la surveillance simultanée de quelques paramètres vitaux. De plus, l'utilisation de plusieurs appareils crée une redondance tant au niveau du matériel et des fonctions qu'à celui des alarmes, et engendre des difficultés sur le plan de l'utilisation globale des appareils, risquant ainsi de dérouter le personnel soignant /LEDL/. Toutefois, l'apparition récente de plusieurs appareils intégrant les principales fonctions de surveillance dans un module unique a permis d'apporter une plus grande cohérence donc une plus grande efficacité dans l'utilisation des appareillages. Ces systèmes permettent l'analyse de divers paramètres, leur mémorisation s'étend sur une période de temps qui peut aller jusqu'à 48 heures, mais ils n'offrent guère de possibilités d'analyses qualitatives des résultats qu'ils exploitent.

Le principe général du projet a été énoncé il y a quelques années déjà /VAS3/. Cependant, comme il s'agissait d'un travail considérable nécessitant des compétences variées, il a été partagé en deux ensembles principaux:

- L'étude des différents appareils indispensables à la saisie in vivo, c'est à dire sur le patient lui-même, et le traitement des informations caractéristiques de son état pathologique au moment de l'observation.

- L'étude de l'équipement intégré, facile à mettre en oeuvre et à utiliser, dont le but est de recueillir et de synthétiser les différents résultats issus du traitement précédent, afin d'apporter une aide au diagnostic et de déclencher des alarmes en cas d'anomalie nécessitant une intervention immédiate du personnel soignant.

Ainsi, l'orientation du projet est envisagée de la manière suivante:

- Cet équipement intégré est destiné à un seul patient
- Outre le monitoring habituel, des fonctions de bilan physiologique et d'aide à l'examen clinique sont incluses dans cet équipement.

Ceci implique:

- La présence d'une console afin de visualiser les résultats des examens sous diverses formes.
- La modularité de la réalisation rendue nécessaire par l'intégration des fonctions physiologiques dans un seul appareil.

Ce mémoire n'a pas pour but de décrire les méthodes à appliquer lors de la réalisation d'un appareil destiné à analyser en temps réel les signaux d'origine physiologiques. De nombreuses études ont déjà été menées sur ce sujet /CHAMB, CHEB, LECE, VAS2, GEOR, MEHA, CONG, OSBO/. L'étude porte sur le regroupement de ces travaux pour les intégrer au sein d'un seul appareil afin de proposer la réalisation d'un appareillage compact et d'utilisation simple.

Ces bases étant posées, le mémoire est structuré en six chapitres définis comme suit:

Le premier chapitre présente quelques méthodes de traitement du signal utilisées pour la réalisation de

ce système, puis certaines notions physiologiques nécessaires à une bonne compréhension de la présentation des résultats.

Le deuxième chapitre décrit les fonctions réalisées par cette machine et introduit la notion de modularité permettant l'extension ultérieure du système au traitement d'autres signaux. Pour cela, les éléments nécessaires à la réalisation d'une instrumentation intégrée sont exposés, notamment l'analyse des architectures matérielles à base de composants micro-informatiques et des systèmes d'exploitation.

Une synthèse du dialogue homme-machine constitue le troisième chapitre. On y démontre la nécessité d'une présentation claire et attrayante des résultats.

Le quatrième chapitre porte sur l'étude de l'architecture du système, la réalisation matérielle et l'implantation logicielle.

L'étude logicielle ainsi que le "mode d'emploi" de l'appareil font l'objet du cinquième chapitre.

Enfin le sixième chapitre décrit les développements futurs auxquels peut conduire un tel système.

I) PRESENTATION DU PROJET

I-1) Cahier des charges

I-1-1) Signaux du bilan

Ce projet doit permettre l'acquisition par une série de modules électroniques, des signaux à visée cardiovasculaire et respiratoire, et leur intégration au sein d'un seul appareil, afin d'apporter une plus grande efficacité, et une simplicité d'utilisation par le personnel soignant. Chaque module autonome effectue l'acquisition et le traitement en temps réel d'un ou plusieurs signaux. L'objectif de ce travail consiste à collecter, synthétiser et représenter les résultats du traitement des signaux physiologiques /DOMI/.

Pour chaque module, le traitement se décompose en trois étapes:

- Détection du signal en formes successives par isolation de cycles.
- Caractérisation de chaque cycle.
- Classification, qui affecte à chaque cycle une situation donnant un sens à l'évènement traité. Cette classification permet d'apporter une aide au diagnostic.

A titre d'exemple, l'électrocardiogramme est une succession d'ondes pseudopériodiques représentant chacune un évènement "cycle cardiaque". On peut ainsi classer ces évènements parmi une liste de situations prédéfinies afin de pouvoir établir un diagnostic.

Le premier module réalise le traitement automatique de l'électrocardiogramme (E.C.G.). Une étude précédente

/LECE/ a permis de mettre au point un algorithme pouvant répondre à deux questions simples:

- Le rythme cardiaque est-il régulier ou irrégulier?
- Si on décele une anomalie, est-elle d'origine auriculaire ou ventriculaire?

Chaque cycle est classé parmi six situations possibles décrites au chapitre I-4.

Le deuxième module permet de traiter trois signaux d'origine respiratoire:

- VE: volume expiré
- P: pression des voies aériennes
- FCO<sub>2</sub>: concentration instantanée en CO<sub>2</sub> expiré

Ce module a fait l'objet de travaux /CHAM/ qui ont conduit à la commercialisation d'un système de surveillance respiratoire. Les signaux traités par ce module sont exposés au chapitre I-5.

Le troisième module /MEHA/ a pour objectif de suivre l'évolution des pressions sanguines P.A.P. ou T.A. en calculant les pressions moyennes diastoliques et systoliques. Ces signaux sont repris au chapitre I-6.

L'analyse de chaque signal fournit des évènements pour élaborer les deux informations suivantes:

- Un vecteur de paramètres caractérisant l'évènement.
- Le résultat d'une classification affectant l'évènement à une situation.

Par exemple, le cahier des charges du traitement local de l'E.C.G. peut être la classification du rythme cardiaque parmi les trois situations suivantes: rythme normal, tachycardie, bradycardie.

Ainsi, chaque module autonome effectue les actions suivantes:

- L'acquisition et le prétraitement du signal.
- La génération des vecteurs de paramètres caractérisant chaque événement.
- La détection des changements de situation.
- La mémorisation du passé récent (quart d'heure en cours) sous la forme d'histogrammes de répartition des paramètres pour chaque situation.

#### I-1-2) Modularité

L'intérêt de cette conception modulaire permet l'adjonction ou le retrait (pour modification) des modules autonomes sans remettre en cause la gestion de l'ensemble du système. Si la modularité matérielle est élémentaire, il n'en va pas de même pour la modularité logicielle. Le logiciel doit être capable de s'adapter aussi facilement que possible à toute modification apportée au système global. Chaque module autonome est une carte à microprocesseur dont il faut exploiter les résultats du traitement. Cette exploitation ne peut se faire que par une autre carte à microprocesseur ayant pour tâche de rassembler le travail des modules autonomes. C'est au niveau de cette carte qu'intervient la modularité logicielle qui doit être conçue pour ne modifier qu'un minimum de logiciel en cas de modification matérielle. Ceci peut être réalisé par simple adjonction d'une PROM sur la carte chargée d'exploiter les résultats d'un nouveau module autonome connecté au système.

#### I-1-3) Fonctions de l'appareil

L'appareil intégré doit remplir les fonctions suivantes:

- Mémoriser le passé du patient sur 24 heures.

- Délivrer sur demande de l'opérateur une image synthétique du bilan. Un tel bilan doit permettre de réaliser deux types d'observations:

- Le présent

- Une fenêtre quelconque du passé: on admet toutefois que la largeur minimale de la fenêtre est de un quart d'heure.

- Permettre à l'opérateur d'établir d'éventuelles corrélations entre les signaux: d'où la nécessité de pouvoir visualiser les différentes situations (sous diverses formes graphiques) reconnues lors de l'analyse de chacun des signaux.

- Générer des alarmes: cet appareil peut connaître à tout instant l'état de chaque signal. Il est donc intéressant de concevoir l'alarme comme le résultat de la situation de l'ensemble des signaux.

Cet exposé succinct des objectifs du projet autorise maintenant l'étude détaillée de la réalisation et du fonctionnement de chacun des modules.

## I-2) Méthode de Calvez

Une méthode de conception s'avère indispensable pour mener à bien ce projet qui s'assimile tout à fait à un système d'observation complexe. Calvez /CAL1, CAL2/ a développé une méthode qui répond parfaitement à la réalisation du projet considéré. L'application de la méthode de Calvez à ce projet a été étudié par Daniel Dupont /DUPO/. Cette méthode se décompose en cinq parties.

### I-2-1) Définition des spécifications

Cette première étape définit les contraintes liées à la réalisation du projet. Elles représentent le cahier des charges de l'ensemble des fonctions que doit réaliser le système, leurs caractéristiques, ainsi que

l'implantation technologique imposée par le processus et le lieu d'implantation.

Ainsi, chaque fonction physiologique est traitée localement selon la séquence suivante:

- Un capteur propre à chaque signal permet de réaliser son acquisition.
- Les faibles niveaux électriques délivrés par les capteurs sont amplifiés.
- Le signal est filtré pour atténuer les bruits parasites externes.
- La conversion analogique-numérique du signal permet d'effectuer un traitement sur ordinateur.
- Le traitement local permet d'extraire les paramètres quantitatifs et qualitatifs relatifs à la morphologie du signal.

Le traitement global collecte les résultats des traitements locaux et en présente une synthèse selon les ordres reçus de l'opérateur.

#### I-2-2) Structure fonctionnelle

Cette étape définit les règles devant être respectées pour parvenir à une solution qui réponde au mieux aux spécifications. Ces règles se résument comme suit:

- Toute application est décomposable en sous-ensembles pour aboutir à une structure modulaire, structurée et hiérarchisée.
- La décomposition de la structure fonctionnelle se poursuit tant qu'il existe des actions non élémentaires.

- Les actions élémentaires à couplage serré sont regroupées afin d'aboutir à une structure sous forme de modules de communication et de messages asynchrones.

Cette notion de module facilite l'étape de l'implantation matérielle.

Pour ce projet, le patient est perçu comme un générateur de signaux physiologiques analysés par le système, et le personnel soignant comme un observateur amené à réagir en fonction des différents résultats issus du traitement des signaux.

Suite aux définitions des spécifications, un moyen de communication avec l'extérieur doit être introduit à ce niveau. Il ressort de ce système trois fonctions distinctes:

- Une unité de traitement local chargée du traitement des signaux.
- Une unité de traitement global chargée de synthétiser l'ensemble des résultats issus du traitement local.
- Une unité de dialogue permettant la communication opérateur/système.

### I-2-3) Description opératoire

Cette étape détaille le comportement interne des actions élémentaires à partir des deux étapes précédentes. Chaque action est caractérisée par son évolution dans le temps et les échanges avec son environnement. Elle est décomposée en deux parties: une partie initialisation et une partie exécution. Un pseudo-langage définit le processus fonctionnel des actions. On distingue deux types d'actions:

- Action temporaire: son activité est liée à la réception d'une donnée (événement) ou d'une combinaison de données extérieures.

Elle possède la structure suivante:

```

cycle
  événement 1: <procédure 1>;
      :           :
      :           :
  événement n: <procédure n>;
end;
```

Cette action peut être dotée de conditions supplémentaires d'évolution afin d'améliorer les possibilités de synchronisation interactive.

```

when
  événement i: <procédure i>;
  message j: <procédure j>;
end;
```

- Action permanente: cette action n'est pas liée à la présence de messages ou d'événements extérieurs. Le processus est décrit comme suit:

```

cycle
  <procédure>;
end;
```

On retrouve ces actions tant au niveau des divers traitements qu'au niveau du dialogue avec le monde extérieur.

Ainsi, en fonction d'un événement du signal physiologique, le traitement local effectue une action qui est fonction de cet événement.

De même au niveau du dialogue: l'appui sur une touche de la console déclenche le processus de test sur cette touche ainsi que l'activation de l'action correspondante.

#### I-2-4) Structure exécutive

Cette étape détermine l'ensemble des ressources et des liens nécessaires au fonctionnement de l'application. Trois constituants principaux permettent la structure exécutive:

##### - Les processeurs:

Propriétés:

- 1) La durée d'exécution d'une action d'un processeur est finie.
- 2) Une signalisation extérieure peut modifier l'activité d'un processeur.
- 3) Un processeur travaille avec les ressources qui lui sont dédiées.
- 4) Un processeur peut activer des sorties.
- 5) Une action élémentaire est assurée par un seul processeur.
- 6) Un processeur peut être le support d'exécution de plusieurs actions, à condition de respecter les contraintes de temps d'exécution (taux d'occupation du processeur  $< 1$ ).

##### - Les mémoires:

- 1) La durée d'accès à une mémoire est finie.
- 2) Les contraintes d'intégrité doivent être garanties.
- 3) Une mémoire détecte si une variable concernée par un accès en lecture ou en écriture lui appartient.

##### - Les liaisons:

- 1) Les liaisons assurent une transmission correcte des messages.
- 2) En cas de demande de plusieurs transmissions simultanément, les conflits d'accès sur la liaison doivent être résolus.

Cette étape est le résultat de la synthèse entre la description opératoire et la structure fonctionnelle.

Le critère de modularité est la spécification majeure de ce projet. Chaque signal est donc traité par une seule carte à microprocesseur. L'adjonction d'autres cartes peut permettre le traitement de signaux supplémentaires. Ce critère est pris en compte dans la réalisation logicielle, afin d'éviter une refonte totale du système en cas d'extension. Le traitement global est assuré lui-même par une seule carte.

#### I-2-5) Réalisation

Cette dernière étape consiste à choisir un matériel approprié à la réalisation, puis à implanter le logiciel dans le matériel.

##### I-2-5-1) Réalisation matérielle

Face à certaines spécifications technologiques imposant des critères de construction (modularité, minimisation du coût, sûreté de fonctionnement), les concepteurs doivent faire un choix quant au matériel à adopter.

Dans le cas de la modularité, la réalisation se limite à l'assemblage de cartes standard et de quelques développements externes liés au contrôle du processus.

Concernant la minimisation du coût, toute la construction matérielle correspondante reste à faire.

La sûreté de fonctionnement est prise en compte au niveau de la structure exécutive, par l'introduction de redondances et de contraintes d'exclusion de certaines actions.

### I-2-5-2) Réalisation logicielle

Elle s'obtient par traduction manuelle de la description opératoire sur la structure exécutive. Cette réalisation implique trois problèmes:

- Partage du processeur entre plusieurs actions, donc fonctionnement du processeur par interruptions.
- Cohérence des variables partagées: il faut garantir l'intégrité des variables et donc, pour cela, définir des techniques d'accès. Dans un contexte temps réel, il faut assurer la remise à jour très rapide des variables pour respecter le contexte.
- Gestion des noeuds de communication: il est nécessaire de faire appel aux techniques que l'on rencontre dans les systèmes d'exploitation pour partager les ressources.
- Prise en compte des événements et des messages: utilisation des instructions "cycle" et "when".

### I-3) Traitement du signal

#### I-3-1) Introduction

Un phénomène temporel est une succession dans le temps d'évènements caractéristiques. L'enregistrement de la figure 1-1 est un tracé standard de l'E.C.G. sur lequel on distingue la succession d'ondes pseudo-périodiques représentant chacune un évènement cycle cardiaque.

Chaque cycle cardiaque est caractérisé par une suite de paramètres dont les plus courants sont notés P Q R S T.

Pour réaliser le traitement de l'électrocardiogramme précédent, on procède à une

conversion analogique-digitale à la fréquence  $T_e$ . Ainsi l'évènement cycle cardiaque apparaît comme un assemblage d'éléments caractérisés par l'amplitude du signal à l'instant d'échantillonnage. A partir de ces éléments, il est alors possible de reconnaître le début et la fin du cycle cardiaque par détection de l'onde P et T.

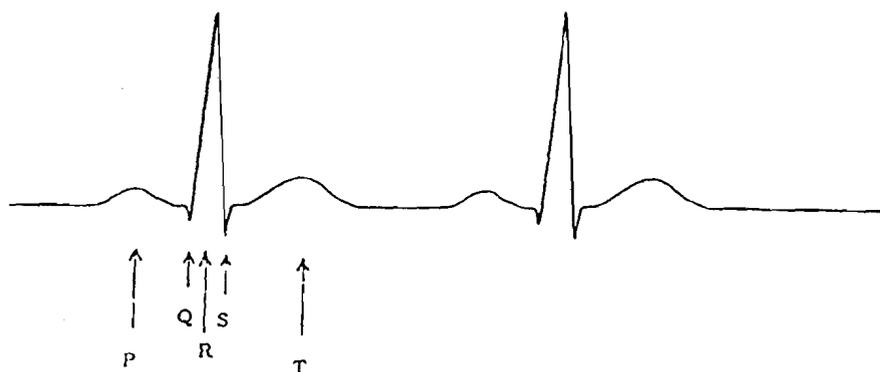


Figure 1-1

On peut donc construire un ensemble de notions élémentaires indispensables à l'analyse des phénomènes temporels (objet, date, durée)/LECE/.

### I-3-2) Analyse du signal

L'observation d'un phénomène temporel se situe à divers niveaux d'interprétation. Chaque niveau est caractérisé par un type d'évènement dont l'assemblage permet d'accéder à un niveau d'interprétation supérieur. La durée des évènements observés est d'autant plus longue que l'on se situe à un plus haut niveau d'observation /VAS3/.

L'analyse du signal a pour but de faciliter la compréhension du processus en passant d'un espace de représentation du signal analogique à un espace d'interprétation par classification automatique d'évènements sous différentes formes (histogrammes, courbes d'évolution, tableaux de coefficients). Le

traitement fait appel à un ensemble de données (modèle, prototype) établies à priori ou mises à jour en permanence au cours de l'étude.

#### I-3-2-1) Mémorisation et traitement

La mémorisation permet la sauvegarde d'une partie utile des informations relatives aux différents niveaux d'interprétation de la phase de travail. Les données spatiales constituent l'amplitude des grandeurs analogiques représentant le signal, et les données temporelles les écarts séparant les événements élémentaires spécifiques, exprimés en nombre de périodes d'échantillonnage (temps).

En dehors de ces données qui sont des résultats exploités par l'utilisateur, on peut être amené à manipuler des indicateurs pour le contrôle des diverses opérations dépendant de la situation dans laquelle on se trouve, et des coefficients qui agissent sur les résultats issus du traitement.

On est donc conduit à différencier deux manières de stockage des données.

a)-L'information est figée définitivement: cas de la mémorisation des résultats.

b)-L'information est remise à jour en permanence: cas d'indicateurs, de paramètres ou de résultats intermédiaires de calcul.

le traitement correspond à la programmation des méthodes adoptées pour effectuer l'analyse du signal /LABA/. Un traitement en temps réel étant imposé, chaque prise d'échantillon doit être suivie d'un traitement. Dans le cas le plus favorable, le traitement est synchronisé sur la prise d'échantillon, ce qui veut dire que la durée du traitement n'excède pas la période d'échantillonnage. Lorsque le traitement est plus approfondi, il est possible que la durée de l'analyse déborde sur la période d'échantillonnage.

Ceci implique que tout dépassement doit être compensé par des intervalles de temps laissés libres par des traitements plus courts. Dans ce cas, tous les échantillons sont bien pris en compte (figure 1-2).

Ce mode de traitement ou mode "désynchronisé" est beaucoup plus souple que le mode "synchronisé". Il faut s'intéresser à la durée moyenne de traitement d'un cycle et non à la durée instantanée du traitement pour savoir s'il y a débordement ou non.

### I-3-2-2) Contraintes temps réel

Le problème du mode "désynchronisé" consiste à connaître la durée moyenne de plusieurs traitements consécutifs sur une fenêtre donnée. Un algorithme bien choisi permettra de ne pas prendre en compte les échantillons non caractéristiques d'un cycle pour compenser la durée de traitement liée aux échantillons représentatifs du cycle (début et fin de cycle, perturbations du cycle).

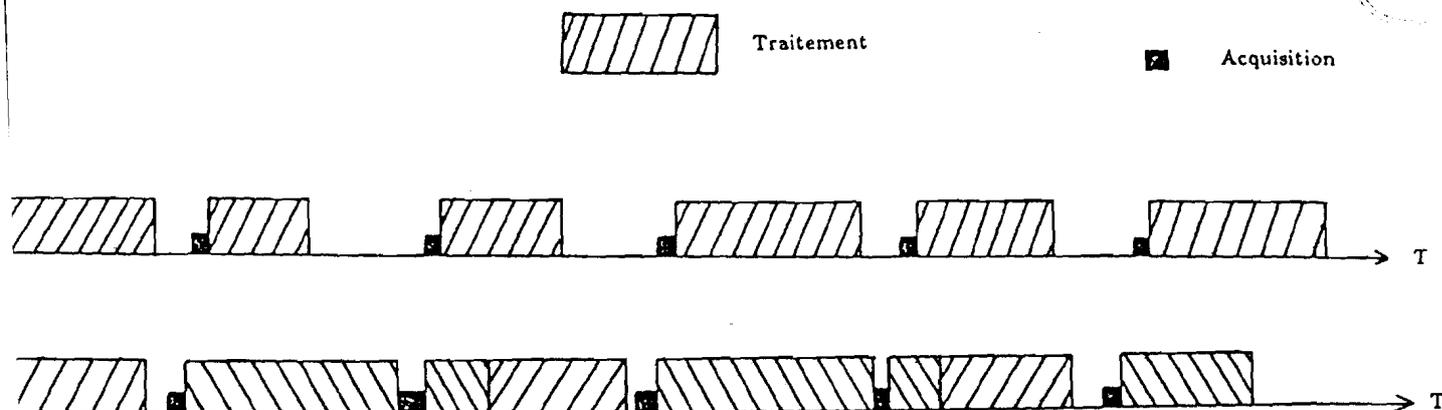


Figure 1-2

Par exemple, le signal issu d'un capnigraphe donnant la concentration en  $\text{CO}_2$  du gaz expiré permet deux sortes de traitement (figure 1-3).

a)-Une fonction de surveillance limitée dans la détection de deux seuils pré-réglés par un médecin. Le

dépassement de ces seuils provoque une alarme soit par concentration de  $\text{CO}_2$  trop importante (seuil haut), soit par franchissement du seuil bas.

b)-L'autre fonction revient à analyser globalement le cycle de manière à réaliser une aide au diagnostic. Pour cela, il faut pouvoir établir une classification automatique des différents événements survenus lors de chaque cycle et détecter les éventuels phénomènes d'accrochage, d'asynchronisme ou d'instabilité.

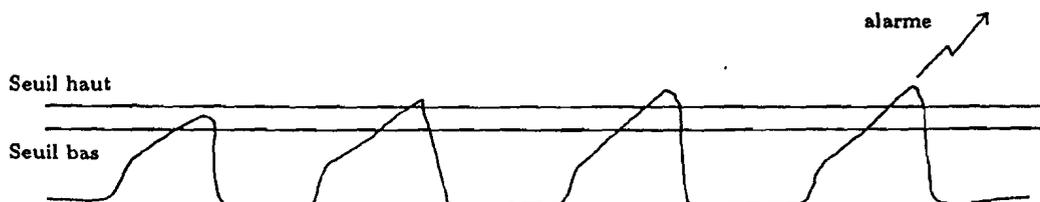


Figure 1-3

### I-3-3) Conclusion

Les méthodes de traitement d'un signal d'origine biologique à caractère cyclique sont celles qui facilitent l'examen du rythme, des variations de paramètres, des répétitions... Il est donc intéressant d'assimiler le signal observé à une suite d'événements selon l'interprétation recherchée. Ainsi, l'analyse du signal s'effectue par plusieurs phases spécifiques: validation, paramétrisation, classification.

## I-4) Application au traitement de l'E.C.G.

### I-4-1) Introduction

L'objectif à atteindre pour le premier module de traitement de l'appareil à réaliser est l'analyse de l'électrocardiogramme dans un contexte d'urgence

I-4-2) Description de l'électrocardiogramme

Le principe de base de l'électrocardiogramme consiste à enregistrer l'activité électrique du coeur au moyen d'électrodes collées sur la peau en regard de la zone cardiaque. Chaque impulsion émise par le coeur est transmise après traitement, soit à un stylet encre qui reproduit un tracé de cette activité sur une bande de papier, soit sur un écran cathodique. Chaque élément du tracé a une spécification bien précise et donne toujours des renseignements précieux sur le fonctionnement du coeur en complément de l'examen clinique. Il permet de renseigner sur le rythme cardiaque, sa fréquence, sa régularité, la taille des cavités cardiaques, la conduction des influx nerveux le long des parois du coeur, et d'éventuels infarctus /HEW4/.

Sur le tracé de la figure 1-4 on distingue cinq ondes appelées P Q R S T qui s'incrivent soit au dessus (ondes positives P R T) soit en dessous (ondes négatives Q S) de la ligne isoélectrique correspondant à l'absence d'activité électrique (diastole).

-L'onde P correspond à la conduction dans les oreillettes donc à la propagation de l'onde d'excitation provenant du noeud sinusal.

-L'intervalle P Q représente la jonction auriculo-ventriculaire.

-Le complexe Q R S correspond à la conduction dans les ventricules (propagation de l'onde d'excitation en aval du noeud auriculo-ventriculaire dans les deux branches du faisceau de His).

-L'onde T représente la repolarisation de la surface ventriculaire.

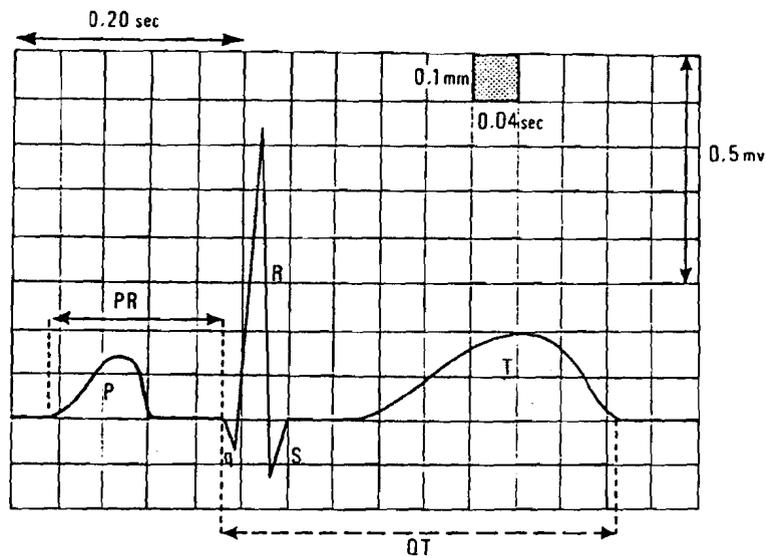


Figure 1-4

I-4-3) Troubles du rythme

On entend par troubles du rythme toute modification du fonctionnement normal du coeur. L'activité électrique du coeur prend naissance au noeud sinusal, chemine par les oreillettes, gagne le noeud de Tawara, puis parcourt le faisceau de His pour déclencher la contraction musculaire.

Toute modification dans cette excitation sinusale ou dans cette transmission jusqu'au ventricule va provoquer un trouble du rythme. On distingue ainsi trois grands ordres de troubles /EGIC, BLON/.

- Les dérèglements du rythme sinusal
- Les rythmes ectopiques passifs et actifs
- Les troubles de la conduction

Les dérèglements du rythme sinusal représentent les troubles les plus fréquents et les plus banaux.

Une tachycardie sinusale est une simple accélération du rythme cardiaque, une bradycardie sinusale un ralentissement du cycle normal. Dans les deux cas, la morphologie du complexe reste normale.

Il y a rythme ectopique passif lorsque la commande sinusale étant devenue défaillante, la commande est alors prise en charge par un foyer d'excitation situé en aval, soit le noeud de Tawara, ou si celui-ci est également défaillant, le faisceau de His.

Il y a rythme ectopique actif lorsque des rythmes anormaux naissent en dehors des voies de conduction normales. Dans ce cas, on parle de fibrillation ou flutter.

Dans les troubles de la conduction, c'est la transmission de l'influx qui se fait de façon anormale soit entre le noeud sinusal et oreillettes (bloc sino-auriculaire), soit entre les oreillettes et les ventricules (bloc auriculo-ventriculaire).

Cet exposé simplifié des troubles du rythme cardiaque est suffisant pour introduire le cahier des charges. On se propose donc de localiser et de classifier six types de paramètres, plus la valeur de la fréquence cardiaque:

- Rythme régulier sinusal (RCF)
- Arythmie à complexe fin (EF)
- Rythme régulier à complexe large (RCL)
- Arythmie à complexe large unidirectionnel (EL)
- Arythmie à complexe large polymorphe (LP)
- Indéterminé: problème d'électrodes ou autre (ND)

Cette classification (figure 1-5) correspond aux informations extraites de l'électrocardiogramme détaillées comme suit:

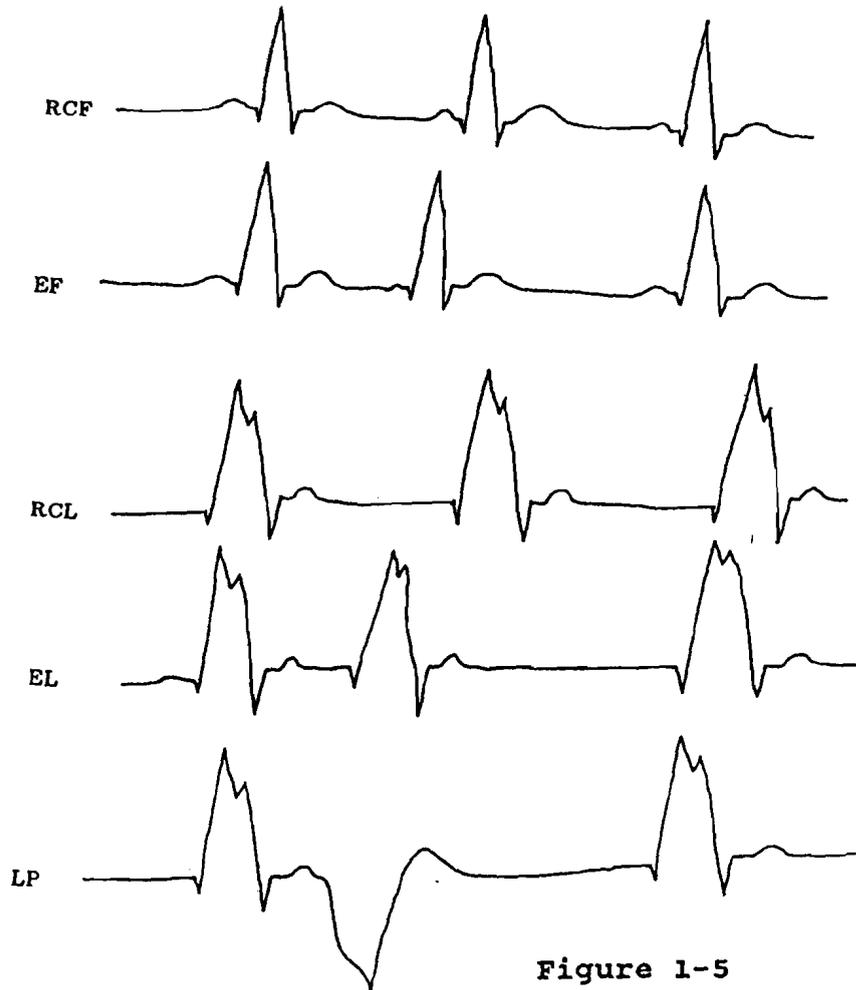
- Rythme régulier sinusal: rythme normal du coeur.
- Arythmie à complexe fin: dans cette classe se trouvent deux possibilités:
  - a) La cause de l'arythmie domine le noeud sinusal et a tendance à exciter prématurément le ventricule.
  - b) Il y a blocage entre le noeud sinusal et le noeud auriculo-ventriculaire si bien que c'est ce dernier qui devient entraîneur. Sa fréquence propre étant plus basse, l'excitation des ventricules se fera en retard.

La déformation du signal est donc caractérisée par une onde P inexistante ou négative, juste avant ou juste après le complexe ventriculaire.

- Rythme régulier à complexe large: le noeud sinusal est l'entraîneur, mais on assiste à des troubles de la conduction caractérisés par des ondes retardées. L'intervalle QS augmente.

- Arythmie à complexe large unidirectionnel: elles naissent d'un ou plusieurs foyers d'activité ectopique. Elles préludent souvent à un trouble du rythme grave: tachycardie ou fibrillation ventriculaire. Elles se caractérisent par une déformation imprévisible du complexe QRS et une augmentation de l'intervalle QS. Une série de cycles cardiaques est toujours soit positive, soit négative de part et d'autre de la ligne isoélectrique.

- Arythmie à complexe large polymorphe: par opposition à l'arythmie à complexe large unidirectionnel, une suite de cycles cardiaques est totalement désordonnée.



#### I-4-4) Traitement

Le signal électrocardiographique est analysé en 2 temps :

Acquisition : le signal est filtré, amplifié et échantillonné à la fréquence de 100 Hz.

Analyse: elle est basée sur la dérivée qui est calculée par la méthode des segmentations /RAJA/.

L'analyse du signal se déroule en trois phases.

-Phase d'apprentissage: après lissage sur 5 points /VAS2/, les différents paramètres caractéristiques d'un cycle sont mesurés puis mémorisés /MURA, VICT/. Les paramètres sont les suivants.

-Largeur moyenne du complexe Q-R-S

-Amplitude moyenne du complexe Q-R-S

-Distance moyenne Q-T

-Distance moyenne R-R

-Test du modèle: cette phase compare les cinq complexes suivants au modèle. En effet, la phase d'apprentissage a très bien pu se dérouler pendant un cycle à complexe anormal. Si la comparaison montre une anomalie, le programme retourne à la phase précédente.

-Phase opérationnelle: chaque cycle est analysé et classé selon les six catégories décrites précédemment. Chaque cycle qui est reconnu R-CF est considéré comme normal et remplace le modèle de référence ce qui permet une adaptabilité du modèle, car l'E.C.G. du patient peut évoluer dans le temps.

Du point de vue visualisation, le traitement génère deux histogrammes par quart d'heure comptabilisant le nombre de cycles, l'un en fonction de la fréquence cardiaque, l'autre en fonction des situations et un graphe de situation.

#### I-4-5) Conclusion

Le traitement de l'E.C.G. constitue le premier module d'analyse destiné à alimenter le système intégré de surveillance des fonctions physiologiques. Les paragraphes suivants décrivent les autres signaux (P.A.P. et respiratoires) et les informations que doit faire ressortir leur traitement pour constituer deux modules supplémentaires destinés à être adjoints au système.

### I-5) Signaux respiratoires

#### I-5-1) Introduction

La respiration pulmonaire est l'une des grandes fonctions qui participe au maintien de l'équilibre du

milieu intérieur de l'individu. Cette fonction peut être gravement perturbée par de multiples causes et nécessiter une intervention extérieure pour assurer l'échange  $O_2/CO_2$  avec l'air ambiant /GOTH/. Les prothèses respiratoires qui permettent d'assurer la ventilation artificielle peuvent fournir des renseignements nécessaires à la surveillance continue des patients.

L'objectif de ce second module est d'analyser les signaux d'origine respiratoire qui sont: le volume expiré, la concentration en gaz carbonique de l'air expiré et la pression des voies aérienne. Pour cela il convient également de classifier en temps réel les cycles respiratoires /GEOR, ITOH/. L'analyse cycle par cycle implique trois types d'actions:

- La détermination du cycle.
- L'extraction de paramètres caractéristiques pour chaque cycle.
- La classification du cycle.

Les paramètres que ce module doit être capable de traiter sont présentés pour chaque signal.

#### I-5-2) Présentation des signaux

L'étude de ce module porte sur le traitement de trois signaux:

- Volume de gaz expiré
- Concentration en  $CO_2$  du gaz expiré
- Mesure de pression

Le signal de volume est fourni par un capteur de débit placé en bout de la tubulure expiratoire (Figure 1-6).

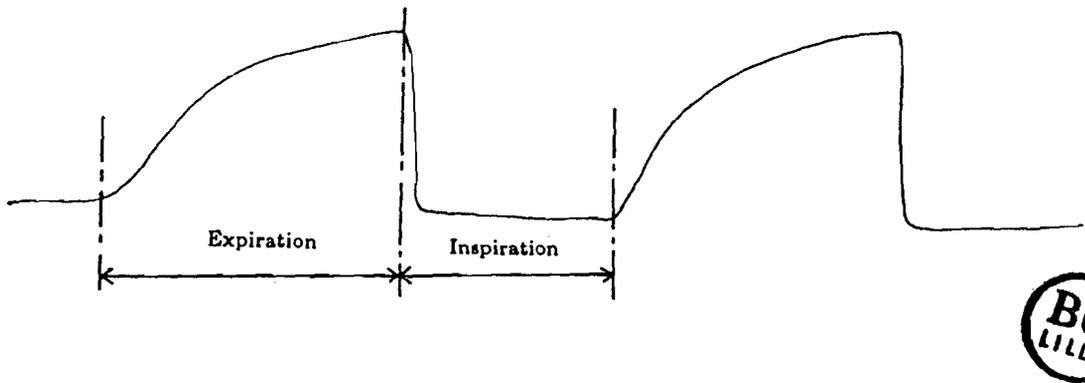


Figure 1-6

L'analyse du signal de  $\text{CO}_2$  s'effectue à l'aide d'un capnigraphe ou d'un spectromètre de masse (Figure 1-7).

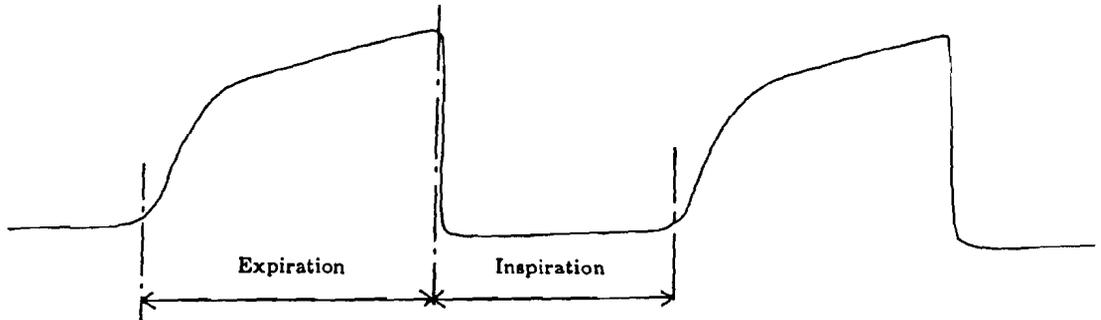


Figure 1-7

L'acquisition du dernier signal est réalisée par un capteur de pression placé au sein même du respirateur (Figure 1-8).

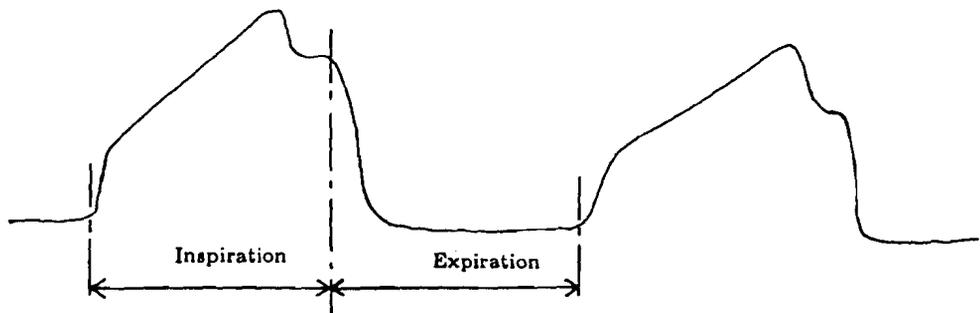


Figure 1-8

Les déformations observées ont pour origine une désynchronisation entre les phases inspiratoires et expiratoires de la personne ventilée et celles imposées par le respirateur. On peut détecter des phénomènes "d'asynchronisme", "d'instabilité" et "d'accrochage" (Figure 1-9).

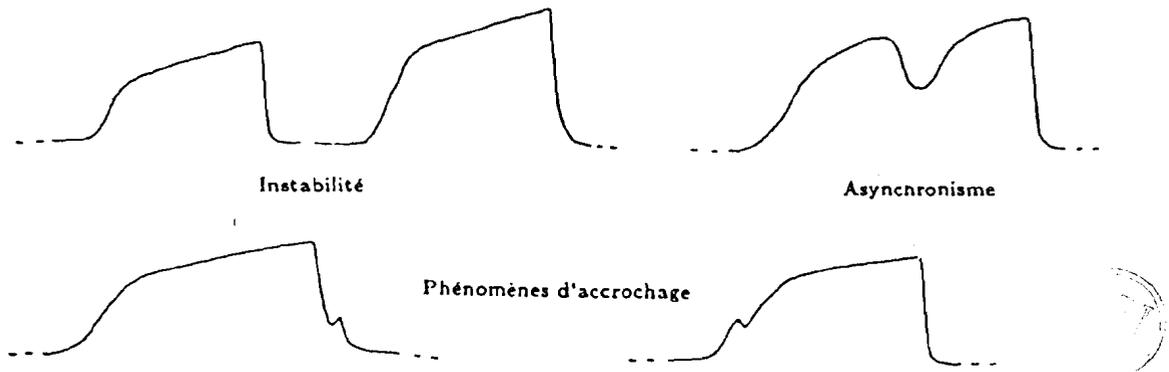


Figure 1-9

#### I-5-3) Calcul des paramètres

Les paramètres caractéristiques de chaque cycle sont obtenus en associant au signal une enveloppe formée de segments de droite séparés par des points de cassure /CHEB/. Ces points se rapportent aux échantillons à partir desquels on observe une nette modification du signal (Figure 1-10). Une méthode pour calculer certains paramètres caractéristiques d'un signal consiste à en extraire ces points de cassure. Les paramètres reposent alors sur l'amplitude d'un point de cassure et l'intervalle de temps le séparant d'un autre.

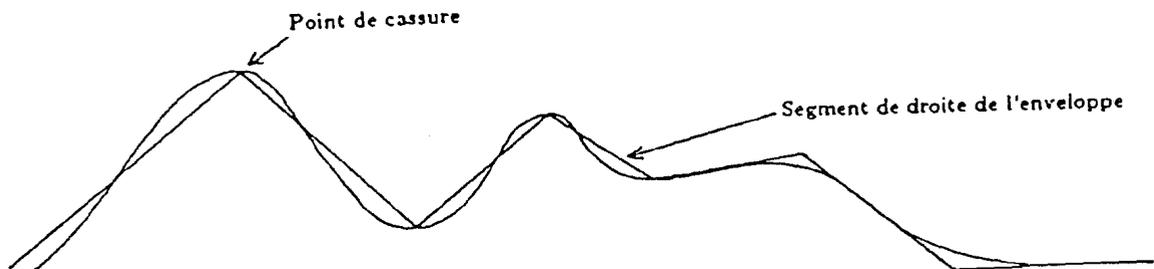
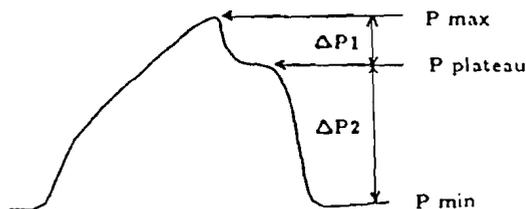


Figure 1-10

I-5-4) Etude des paramètresI-5-4-1) Signal de pression

Le but à atteindre est la classification des cycles respiratoires. L'analyse cycle par cycle impose trois types d'actions: la détermination du cycle, l'extraction des paramètres caractéristiques pour chaque cycle, et leur classification.

Paramètre P+

Le paramètre P+ donne la pression maximum atteinte dans les voies aériennes.

- Aspect qualitatif

- E0: Absence
- E1: Stabilité
- E2: Augmentation
- E3: Diminution

La détection d'une augmentation ou d'une diminution est fonction d'une variation du paramètre de plus ou moins 5 cm de pression d'H<sub>2</sub>O par rapport aux conditions initiales. Cette valeur est considérée comme révélatrice d'un changement de situation.

- Aspect quantitatif:

Du point de vue quantitatif le traitement génère un histogramme du nombre de cycle en fonction des valeurs de P+ variant de 0 à 50 Cm d'H<sub>2</sub>O par intervalle de dix.

Paramètre P-

Le paramètre P-, donne la pression dans les poumons au cours de l'expiration.

- Aspect qualitatif

- E0: Diminution
- E1: Stabilité
- E2: Augmentation

Pour détecter une augmentation ou une diminution, il faut une variation du paramètre de plus ou moins 2 cm H<sub>2</sub>O par rapport aux conditions initiales.

- Aspect quantitatif: quatre valeurs de P-, variant de 0 à 20 CM d'H<sub>2</sub>O par intervalle de cinq.

Paramètre  $\Delta P1$

L'obtention des paramètres  $\Delta P1$  et  $\Delta P2$  n'est possible que si le mode de ventilation impose un plateau de fin d'insufflation.  $\Delta P1$  est un reflet des résistances pulmonaires.

- Aspect qualitatif

- E0: Diminution
- E1: Stabilité (C.I.)
- E2: Augmentation
- E3: Non calculé

Pour détecter une augmentation ou une diminution, il faut une variation du paramètre de plus ou moins 2 cm H<sub>2</sub>O par rapport aux conditions initiales.

- Aspect quantitatif: quatre valeurs variant de 0 à 20 CM d'H<sub>2</sub>O par intervalle de cinq.

### Paramètre $\Delta P_2$

Ce paramètre est primordial pour le calcul de la compliance statique:

$$Cst = \Delta V / \Delta P, \text{ avec } \Delta P = \Delta P(P \text{ plateau} - P \text{ min})$$

- Aspect qualitatif

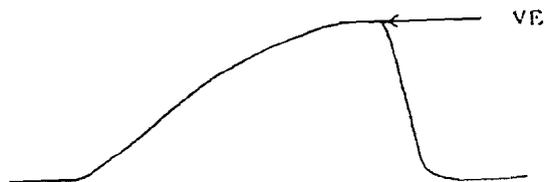
- E0: Diminution
- E1: Stabilité (C.I.)
- E2: Augmentation
- E3: Non calculé

Pour détecter une augmentation ou une diminution, il faut une variation du paramètre de plus ou moins 2 cm H<sub>2</sub>O par rapport aux conditions initiales.

- Aspect quantitatif: trois valeurs variant de 0 à 30 CM d'H<sub>2</sub>O par intervalle de dix.

### I-5-4-2) Signal de volume

Ce signal permet de connaître la valeur du volume courant (VT).



- Aspect qualitatif

E0: Stabilité (C.I.)

E1: Instabilité

E2: Augmentation

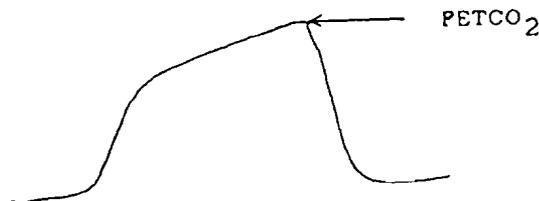
E3: Diminution

Le passage dans l'état E2 ou E3 implique une variation de volume de plus ou moins 10% par rapport à l'état stable E0. 10% est une valeur représentative d'un changement de situation par rapport à la valeur moyenne de ventilation (800 mL).

- Aspect quantitatif: huit valeurs variant de 0 à 800 mL par intervalle de cent.

I-5-4-3) Signal de CO<sub>2</sub>

Ce signal permet de contrôler la teneur en CO<sub>2</sub> de l'air expiré.



- Aspect qualitatif

E0: Absence

E1: Stabilité (C.I.)

E2: Instabilité

E3: Augmentation

E4: Diminution

Le passage dans l'état E3 ou E4 implique une variation de plus ou moins 5% par rapport à l'état stable. 5% est une valeur représentative d'un changement de situation par rapport à la valeur max PETCO<sub>2</sub>.

- Aspect quantitatif: six valeurs variant de 0 à 60 mmHg par intervalle de dix.

I-5-4-4) Paramètres calculés

Fréquence

- Aspect qualitatif

E0: Stabilité (C.I.)

E2: Augmentation

E3: Diminution

Un changement d'état correspond à une variation de plus ou moins 2 cycles/mn.

- Aspect quantitatif: cinq valeurs variant de 0 à 50 cycles/mn par intervalle de dix.

Ductance ventilatoire (DuV)

La ductance ventilatoire est donnée par le rapport  $PICO_2 - PECO_2 / PICO_2 - PAECO_2$ , et permet de révéler l'existence de troubles ventilatoires /LACO/. Le rapport doit toujours être interprété en fonction de la fréquence ventilatoire. La valeur moyenne se situe autour de 0,7.

$PICO_2$  = valeur moyenne inspirée

$PECO_2$  = valeur moyenne expirée

$PAECO_2$  = valeur moyenne alvéolaire

- Aspect qualitatif

E0: Stabilité (C.I.)

E1: Augmentation

E2: Diminution

Un changement d'état correspond à une variation de plus ou moins 0,1 par rapport à l'état stable.

- Aspect quantitatif: huit valeurs variant de 1 à 0,9 par intervalle de 0,1.

Quantité de CO<sub>2</sub> expiré (VCO<sub>2</sub>)

VCO<sub>2</sub> est la quantité de gaz carbonique expiré. VCO<sub>2</sub> est mesurable en fin de cycle. Ce paramètre est calculé à partir des données fournies par un analyseur infra-rouge branché au niveau buccal en utilisant la relation:  $VCO_2 = VE * FECO_2$ .

- Aspect qualitatif

E0: Stabilité (C.I.)

E1: Augmentation

E2: Diminution

Une variation de plus ou moins 10% par rapport aux conditions initiales implique un changement d'état.

- Aspect quantitatif: cinq valeurs variant de 0 à 250 mV par intervalle de cinquante.

Compliance statique (Cst)

La compliance statique est la capacité pour un poumon de se distendre en fonction d'un volume d'air emmagasiné. Elle se détermine en mesurant la variation  $\Delta VE / \Delta P_2 / \text{TREM} /$ .

- Aspect qualitatif

E0: Non calculé

E1: Stabilité (C.I.)

E2: Augmentation

E3: Diminution

Un changement d'état correspond à une variation de plus ou moins 10% par rapport aux conditions initiales. Cette valeur est fonction de l'évolution du malade sur plusieurs jours.

- Aspect quantitatif: quatre valeurs variant de 0 à 60 mL/cm H<sub>2</sub>O par intervalle de quinze.

Inspiration / expiration (I/E)

I/E représente le rapport du temps d'inspiration sur le temps d'expiration.

- Aspect qualitatif

E0: Stabilité (C.I.)

E1: Augmentation

E2: Diminution

Toute variation de plus ou moins 0.25 implique un changement d'état.

- Aspect quantitatif: cinq valeurs variant de 0 à 1 par intervalle de 0,25.

$\Delta(ET-A)$

Ce paramètre permet d'avoir la connaissance du gradient alvéolo-expiré en CO<sub>2</sub> afin de déterminer la présence d'un espace mort alvéolaire.

- Aspect qualitatif

E0: Stabilité (C.I.)

E1: Augmentation

E2: Diminution

Une variation de plus ou moins 5 mmHg par rapport à l'état stable détermine le passage dans l'état E1 ou E2.

- Aspect quantitatif: cinq valeurs variant de 0 à 10 mmHg par intervalle de deux.

## I-6) Exploration des pressions cardiovasculaires

### I-6-1) Rappels physiologiques

A chacune de ses pulsations, le coeur propulse une certaine quantité de sang qui progresse dans les vaisseaux par des contractions rythmées. Ces contractions comportent deux phases successives: la diastole (remplissage de sang des ventricules) et la systole (contraction ventriculaire).

La diastole correspond à l'ouverture des valvules auriculo-ventriculaires provoquant le passage du sang des oreillettes vers les ventricules.

La systole correspond à l'ouverture des valvules sigmoïdes et l'éjection du sang des ventricules vers les artères.

### I-6-2) Capteurs

Les méthodes de mesure non invasives de la pression artérielle telles que la prise de la tension, l'oscillométrie ou l'effet Doppler n'apportent pas entière satisfaction pour la mesure en continu. La mesure de la T.A. par méthode sanglante est plus appropriée et permet d'effectuer la prise des gaz du sang simultanément.

La mesure de la pression artérielle pulmonaire donne des renseignements complémentaires. La sonde utilisée pour mener à bien cette étude est un cathéter flottant. Cette sonde est introduite généralement par la veine jugulaire et est avancée jusqu'à l'oreillette droite. Les pressions qu'elle transmet donnent des renseignements quant à sa position. La surveillance peut commencer lorsque le cathéter a atteint le point d'insertion de l'artère pulmonaire /GAUD/.

I-6-3) Morphologie

Le capteur transmet un signal qui se présente comme une succession de cycles se composant du couple diastolique-systolique (Figure 1-11).

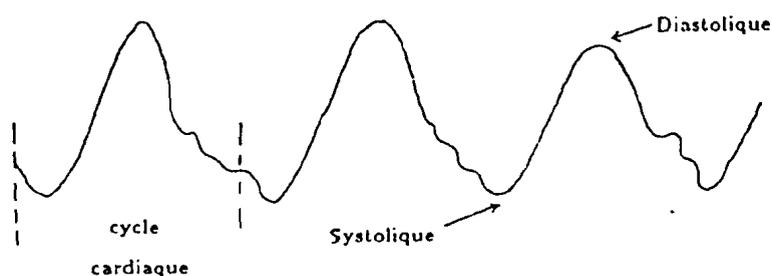


Figure 1-11

Ce signal peut être soumis à d'importantes variations dues notamment aux sautes de cathéter responsables d'artéfacts à fortes amplitudes ainsi qu'aux mouvements respiratoires du patient (Figure 1-12).

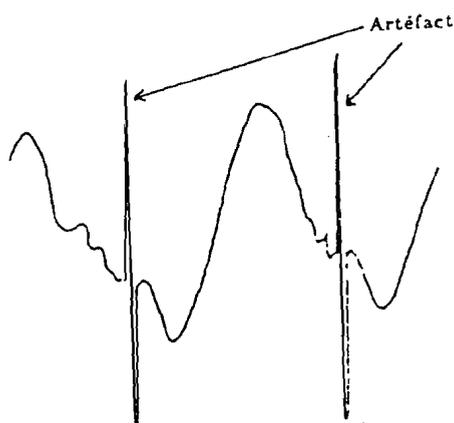


Figure 1-12

I-6-4) But du traitement

Le traitement effectué en temps réel consiste à détecter les couples 'diastolique-systolique' et à en extraire les informations suivantes:

- Valeur moyenne du signal sur plusieurs cycles.
- Valeur de la pression systolique.
- Valeur de la pression diastolique.

Le traitement doit classer les cycles détectés parmi différentes catégories.

P.A.P.

Au cours du temps, le signal peut se trouver dans l'un des quatre états suivants (Aspect qualitatif):

- Normal (cycles normaux).
- Phénomène de résonance (présence d'artéfacts).
- Non interprétable (signal inutilisable).
- Instable (la situation varie plus de trois fois par minute).

Aspect quantitatif:

Cinq fourchettes ont été fixées sur la valeur de la P.A.P. moyenne (mmHg).

- < 10
- 10 à 20
- 20 à 30
- 30 à 40
- > 40

T.A.

La tension artérielle est interprétée en fonction de la différence entre la T.A systolique (TAS) et la T.A. diastolique (TAD). On distingue trois états

définis par les relations suivantes (aspect qualitatif):

- $TAS - TAD > 2/3 TAS$
- $TAS - TAD < 1/3 TAS$
- $1/3 TAS < TAS - TAD < 2/3 TAS$

Aspect quantitatif:

On définit également cinq fourchettes sur la valeur de la tension systolique (mmHg).

- $< 50$  (Collapsus)
- 50 à 100 (Hypotension)
- 100 à 150 (Normotension)
- 150 à 200 (Hypertension moyenne)
- $> 200$  (Hypertension sévère)

#### I-7) Conclusion

L'examen détaillé de l'ensemble des paramètres entraîne maintenant l'étude de la structure fonctionnelle du système.

II) ASPECTS FONCTIONNELSII-1) Structure micro-informatisée

La méthode de Calvez a fait ressortir la nécessité d'utiliser une structure multi-processeurs. Parmi les nombreuses solutions de réalisations existantes /VERJ, FATH/, deux d'entre elles semblent les mieux adaptées à ce projet.

II-1-1) Introduction

Les structures conventionnelles sont centralisées autour d'un seul élément de traitement numérique auquel sont reliées des cartes de mémorisation ou de traitement de l'information.

Le dispositif de traitement numérique est chargé de la gestion de tous les périphériques. Lorsque la densité des échanges atteint un niveau élevé, le temps consacré au transfert des données devient important par rapport au temps consacré au traitement des informations. D'autres structures s'avèrent donc nécessaires pour s'adapter au traitement en temps réel /ROUC, LIU/.

Les descriptions suivantes montrent d'une part les difficultés de conception de structures micro-informatisées, et d'autre part les protocoles d'échanges ainsi que leurs qualités et leurs défauts.

Un ensemble de fonctions élémentaires réalisant un travail spécifique pour exécuter simultanément des tâches ou fonctions élémentaires est beaucoup plus approprié au système dont la réalisation est envisagée.

Chaque ensemble forme une unité fonctionnelle et matérielle capable de réaliser un travail spécifique de manière autonome. Il reçoit la dénomination de processeur d'entrée. Chacun d'entre eux est composé des éléments suivants:

- Microprocesseur
- Mémoire ROM, RAM
- Interface d'Entrée/Sortie
- Convertisseur Analogique/Digital

Le regroupement de plusieurs de ces ensembles respectant certaines règles d'enchaînement et de synchronisation de dialogue forme un système multi-processeurs. Un tel système est la base de cette réalisation.

Différentes méthodes de transmission de données sont envisageables pour réaliser l'interconnection du système. Chaque solution nécessite un matériel approprié, des règles de gestions particulières et du logiciel spécifique.

#### II-1-2) Description d'un BUS

En informatique, le terme de BUS désigne un mode de liaison parallèle entre un ordinateur et ses périphériques. Son rôle est donc d'assurer les communications entre les diverses composantes du système: cartes processeurs, périphériques, cartes ou circuits électroniques. Pour caractériser un bus de la façon la plus complète, il est nécessaire de bien préciser le nombre de fils qu'il comporte, les signaux qui transitent par les fils et les procédures d'échange des informations. La normalisation d'un bus implique donc de connaître son organisation physique, son mode de fonctionnement et les procédures d'échange utilisées. Il est également important de connaître quels types de connecteurs mâles et femelles ont été retenus par le fabricant de bus.

Les périphériques peuvent être soit des dispositifs informatiques classiques (terminal, clavier, lecteur de disquettes, imprimante...), soit des périphériques

associés à un ordinateur pour former une chaîne d'acquisition de mesures, soit d'autres ordinateurs.

D'un point de vue général, un bus doit véhiculer tous les signaux utilisés par le système (données, adresses, contrôle et alimentation) donc associés au matériel, mais également les signaux gérant l'échange des informations associées au logiciel. C'est le microprocesseur, ou l'ordinateur ou le processeur principal, qui est chargé de gérer le bus. Il joue le rôle de contrôleur de bus ou de "maître" par rapport au circuit intégré chargé de contrôler le bus. Les périphériques sont des unités "esclaves" qui peuvent se trouver dans deux états distincts: émetteur ou récepteur d'informations.

A un instant donné, il ne peut y avoir qu'un seul émetteur sur le bus. Dans le cas contraire, il y aurait conflit. De même, il ne saurait y avoir plusieurs contrôleurs de bus en action simultanément.

### II-1-3) Types de liaisons

#### a) liaison par bus unique (figure 2-1)

Tous les processeurs sont connectés à un bus unique /VAS2/. Le contrôle du bus n'est assuré que par un seul processeur à la fois. L'accès du bus est partagé temporellement entre les différents processeurs selon une procédure d'allocation déterminée par le contrôle des communications.

Le processeur maître gère l'ordonnancement des tâches exécutées par les processeurs esclaves. De plus, il collecte les résultats fournis et les exploite pour son propre compte. Enfin, il est chargé de gérer l'ensemble des ressources communes à tous les processeurs, mémoires de masse et mémoires tampons.

les avantages d'un tel système sont la simplicité de mise en oeuvre et la possibilité d'extension facile du système.

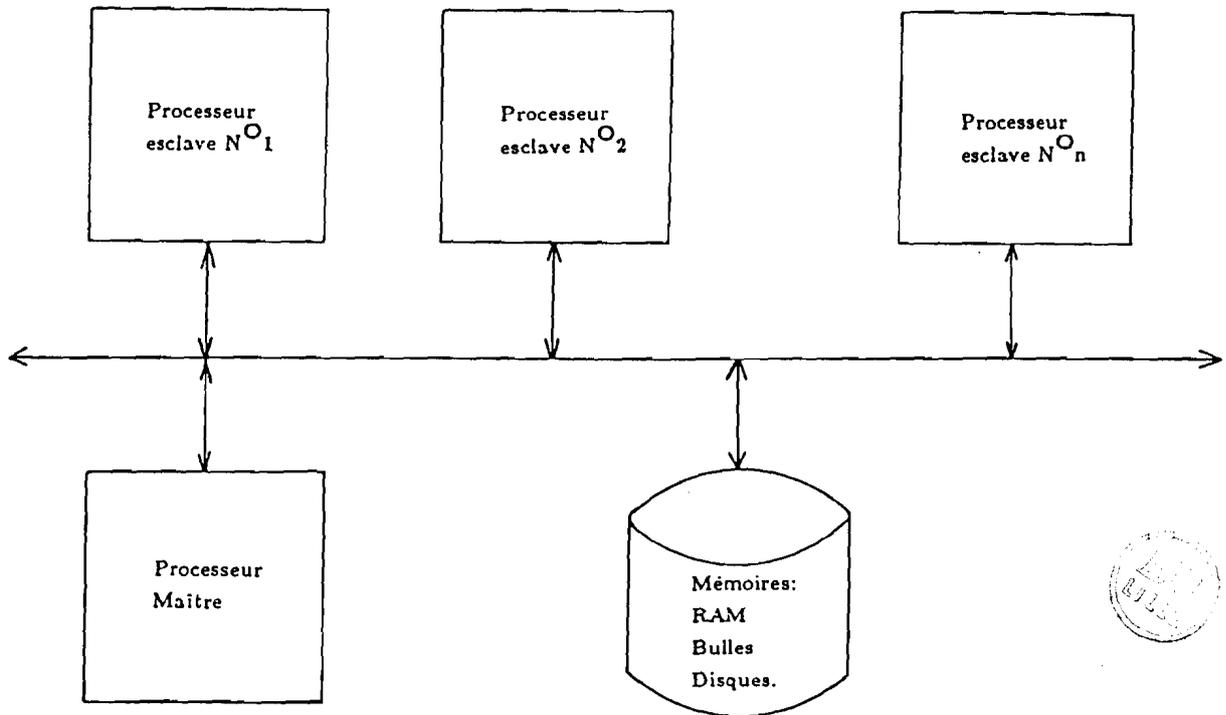


Figure 2-1

## b) structure étoile (figure 2-2)

Le processeur maître centralise l'ensemble des données issues des processeurs esclaves. Il centralise les ressources communes à tous les transferts, mémoires et processeurs, et gère les tâches effectuées par les processeurs esclaves. Le dialogue ne s'effectue qu'avec un seul esclave à la fois.

L'inconvénient majeur de ce système réside dans la gestion trop lourde des transferts de données. Le processeur maître recueille les informations issues des processeurs esclaves et doit les redistribuer aux autres processeurs ou mémoires. De plus le processeur maître doit prendre en charge l'interrogation des processeurs d'entrée, la réception et l'émission de messages.

Pour cette application, le bus unique a été retenu. D'une part, cela permet de mettre en oeuvre un logiciel plus léger, d'autre part, ce type de liaisons convient parfaitement lorsque les distances sont courtes. C'est le cas de notre système. De plus, celui-ci devant être modulaire, l'adjonction de processeurs d'entrée

supplémentaires dans le bus unique ne remet pas en cause toute la gestion de l'appareil.

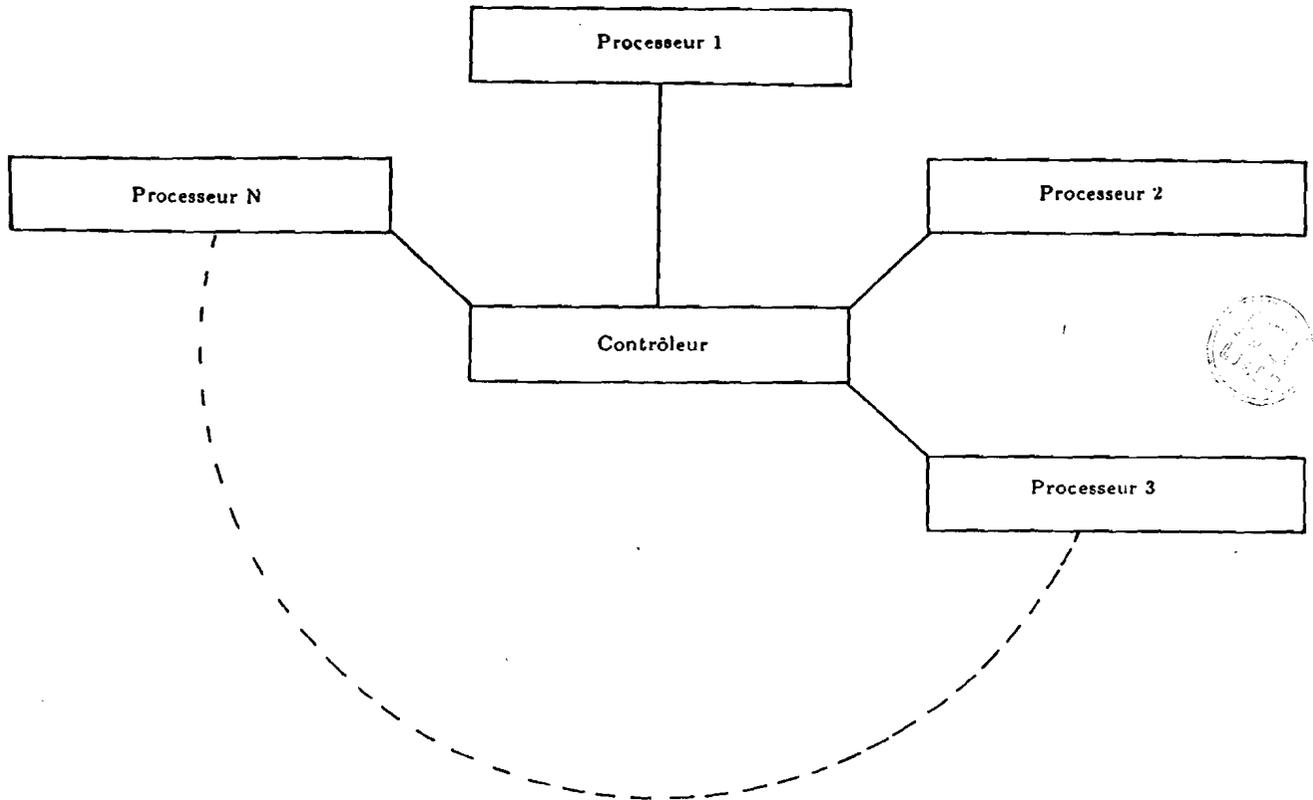


Figure 2-2

#### II-1-4) Choix du matériel

Compte tenu de l'architecture retenue, le choix du matériel s'est porté lors d'une étude précédente /FONT/ sur la marque INTEL qui propose le multibus permettant de relier plusieurs cartes processeurs d'entrée de différentes configurations adaptées au besoin des traitements des signaux. Ainsi, les cartes composant le système ont été choisies dans la famille ISBC.

- Le Multibus a pour référence ISBC 660 /INT4, INT5/.
- les processeurs d'entrée sont des cartes ISBC 80/05 /INT2/.

- Le processeur central est une carte ISBC 80/24 /INT1/.

#### II-1-5) Fonctionnement du Multibus

Le multibus d'INTEL est organisé autour d'un ensemble de quatre vingt six fils répartis de la façon suivante:

- 16 fils de données
- 16 fils d'adresses
- 12 fils de commande
- 10 fils de contrôle et d'adresses
- 8 fils d'interruption
- 12 fils d'alimentation
- 12 fils supplémentaires d'alimentation

La vitesse maximale de transfert des informations est de 5 Mhz.

- Les échanges sur le Multibus

Le Multibus possède une commande totalement décentralisée, ce qui permet la réalisation de structures multiprocesseurs à base d'unités centrales 8 bits, 16 bits, ou même des deux types. La transmission des données et des adresses utilise deux bus distincts et le transfert des données est du type synchrone avec vérification par signal d'acquiescement.

Sauf indication contraire, les signaux de commande du Multibus fonctionnent en logique négative c'est à dire sont actifs au niveau bas. INTEL utilise une notation particulière pour ses signaux, par exemple la commande de lecture mémoire est notée MRDC/ au lieu du traditionnel  $\overline{\text{MRDC}}$ .

La figure 2-3 montre les principes de fonctionnement des signaux pour la lecture et l'écriture sur le bus ISBC.

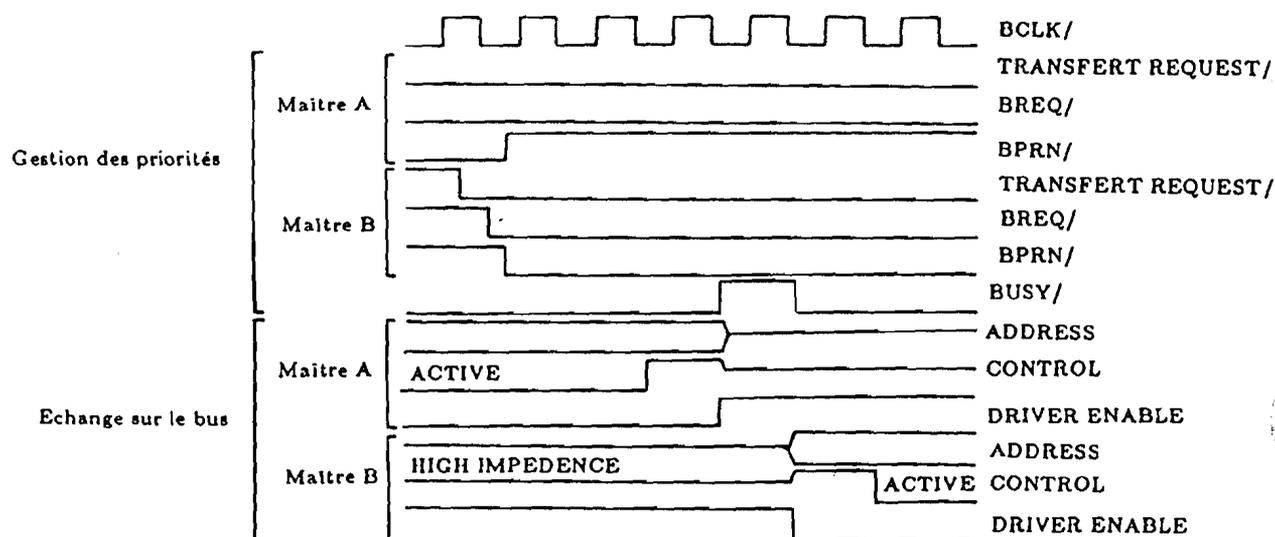


Schéma de résolution parallèle:

le maître B à une priorité supérieure au maître A. Initialement A contrôle le bus et B génère une demande d'occupation.

Figure 2-3

Le multibus ne connaît que des processeurs "maîtres" ou "esclaves" de sorte que tous les échanges se passent entre maîtres et esclaves sous le contrôle du maître. Une temporisation appelée "Time-out" surveille la ligne d'acquiescement XACK/ sur le processeur maître, ce qui permet de libérer le processeur après un certain délai en cas de non réponse /INT1/.

#### II-1-6) Commande décentralisée du Multibus

Plusieurs modules maîtres pouvant prendre simultanément le contrôle du bus, des moyens existent pour éviter des conflits ou même un inter-blocage du Multibus. D'une part, chaque carte comporte ses propres ressources en mémoires et E/S. Ainsi, un processeur maître ne demandera l'accès au multibus que si la ressource désirée est à l'extérieur, ce qui réduit considérablement les transferts d'informations. De plus, seules les ressources communes sont accessibles par les processeurs maîtres. Les ressources résidentes sont protégées contre les accès extérieurs. D'autre

part, chaque carte au standard Multibus comporte des circuits intégrés assurant la gestion des priorités selon deux méthodes:

- La gestion série: chaque maître inhibe le maître de priorité inférieure en accédant au Multibus par l'intermédiaire des lignes BPRN/ (bus priority input) et BPRO/ (bus priority output). Ce qui signifie que la priorité d'un module est déterminée par son emplacement physique dans le connecteur "fond de panier". Cependant, il ne peut y avoir que trois modules maîtres avec ce type de liaisons. Ceci est dû au temps de propagation du signal BPRO/.

- La gestion parallèle: la priorité de chaque maître est déterminée par des circuits logiques de décodage additionnel. Dans ce cas, la ligne BREQ/ (bus request) remplace la sortie BPRO/ de la gestion précédente. La sortie BPRO/ doit donc être inhibée. Lorsqu'un module maître requiert le Multibus, le contrôleur de bus envoie un signal BREQ/ au gestionnaire de priorité qui acquitte en renvoyant BPRN/. Si BPRN/ est vrai, il indique au maître qu'il peut prendre le contrôle du bus car aucun maître de priorité supérieure ne l'utilise. Si BPRN/ est faux, cela signifie que le bus est occupé par un maître de priorité supérieure, donc indisponible. Toutefois, BREQ/ reste vrai jusqu'à acquittement de BPRN/, c'est à dire libération du bus. Cette séquence s'effectue conjointement avec le signal BUSY/ qui est faux lorsque le bus n'est pas utilisé.

Le signal CBRQ/ informe le module maître ayant possession du bus qu'un autre maître de priorité supérieure désire l'utiliser. Dans ce cas, le maître propriétaire termine le transfert de ses données avant de rendre la main.

A noter que le signal BCLK/ (bus clock) synchronise les signaux de contrôle du bus.

Ce mode de gestion permet de contrôler jusqu'à seize modules maîtres. L'encodeur de priorité est un circuit intégré 74128 de TI et le décodeur de priorité un 8205 de INTEL. La priorité des modules est établie par la connexion de leur ligne sur le 74128. L'entrée sept a la plus haute priorité tandis que l'entrée zéro a la plus faible.

#### I-2-7) Conclusion

Le multibus SBC ou bus IEEE 796 possède bien d'autres possibilités; néanmoins certaines ne sont pas exploitées par cette configuration:

- Inhibition d'un bloc de mémoire vive ou morte au profit d'une autre zone mémoire différente ou de même type.

- Possibilité d'interruptions vectorisées.

- Possibilité d'exploiter simultanément la présence de cartes maîtres 8 ou 16 bits.

- Sauvegarde de l'état du système en cas de coupure secteur.

La modularité impose donc une configuration en mode de gestion parallèle du multibus. Ainsi huit cartes maîtres peuvent échanger leurs données.

#### II-2) Mémoire de masse

Une mémoire de masse est nécessaire pour le stockage des résultats du traitement des signaux. Il a donc fallu choisir entre les systèmes englobant les support magnétiques (bande ou disque) et les mémoires à bulles.

Actuellement, les supports magnétiques possèdent une capacité de stockage élevée (>50 Moctets). Néanmoins, leur mise en oeuvre nécessite des mouvements mécaniques intenses d'entraînement des disques et des

têtes de lecture, d'où un temps d'accès à l'information assez long.

Les mémoires à bulles n'offrent qu'une capacité de stockage de 512 Koctets en moyenne mais ne nécessitent aucun mouvement mécanique, donc aucun entretien. De plus, le temps d'accès à l'information est équivalent, voir supérieur à celui des mémoires classiques.

Cette technologie paraît plus appropriée que la précédente. C'est celle qui a été retenue pour cette application. Quant à sa capacité, un rapide calcul vérifie que la taille envisagée de 512 K est largement supérieure au besoin du système. A titre d'exemple, si l'on admet que le rythme cardiaque est de 80 cycles/mn, ceci représente 1200 événements par quart d'heure. Si l'on veut extraire huit paramètres de chaque cycle (donc huit histogrammes) pour chacun desquels seize fourchettes de rythme suffisent largement, la capacité mémoire d'un quart d'heure nécessite:

$$8 * 16 * 2 = 256 \text{ octets}$$

Paramètres Fourchettes Octets/fourchette

et donc 24 Koctets pour 24 heures de mémorisation. Pour le traitement de trois signaux d'origine physiologique, la capacité moyenne de stockage est à peu près égale à 75 Koctets.

## II-3) Système d'exploitation

### II-3-1) Introduction

Un système multi-processeurs implique la notion de système d'exploitation. Ce paragraphe est une introduction aux mécanismes de gestion des processus et des ressources ainsi que la communication entre tâches et périphériques associés à un, voir plusieurs processus. Ces mécanismes sont mis en oeuvre dans le chapitre V consacré à la réalisation.

Un système d'exploitation est un outil logiciel servant à gérer les ressources matérielles de l'ordinateur. Il transforme une machine "réelle" (assemblage de composants électroniques, connecteurs, fils...) en une machine "virtuelle" plus intelligente pour être utilisée avec un maximum de facilité.

Par définition, le système d'exploitation assure un système d'interface entre l'utilisateur et la machine avec son architecture matérielle. De ce fait, le rôle du système d'exploitation peut être abordé de deux points de vue opposés.

- La première approche consiste à se placer du côté utilisateur. Le système d'exploitation est mis en oeuvre à travers un langage de commande de quelques mots clefs, composés à partir d'un clavier.

- La seconde approche est relative à la façon dont sont exécutées les commandes. Ceci implique la résolution des problèmes concernant la gestion mémoire, des entrées/sorties notamment clavier et écran.

### II-3-2) Modèle d'un système d'exploitation

Le concept de système d'exploitation ne prend tout son sens qu'avec les systèmes disposant de puissants moyens de stockage des informations (disques souples ou rigides, mémoires à bulles). Un tel outil est décomposé selon la figure 2-4.

Les différentes fonctions du système d'exploitation sont organisées de façon hiérarchique selon leur complexité et leur niveau d'abstraction. Chaque niveau gère un ensemble de ressources matérielles ou logicielles /LIST/.

Le premier niveau comporte la gestion des éléments matériels.

Le deuxième niveau comprend l'ensemble des instructions processeur (interruptions, piles, gestion de sous-programmes) et les interruptions système. Le processeur peut ainsi sauvegarder l'état du système pour exécuter une autre tâche généralement plus prioritaire telle que réception de messages issus de divers périphériques.

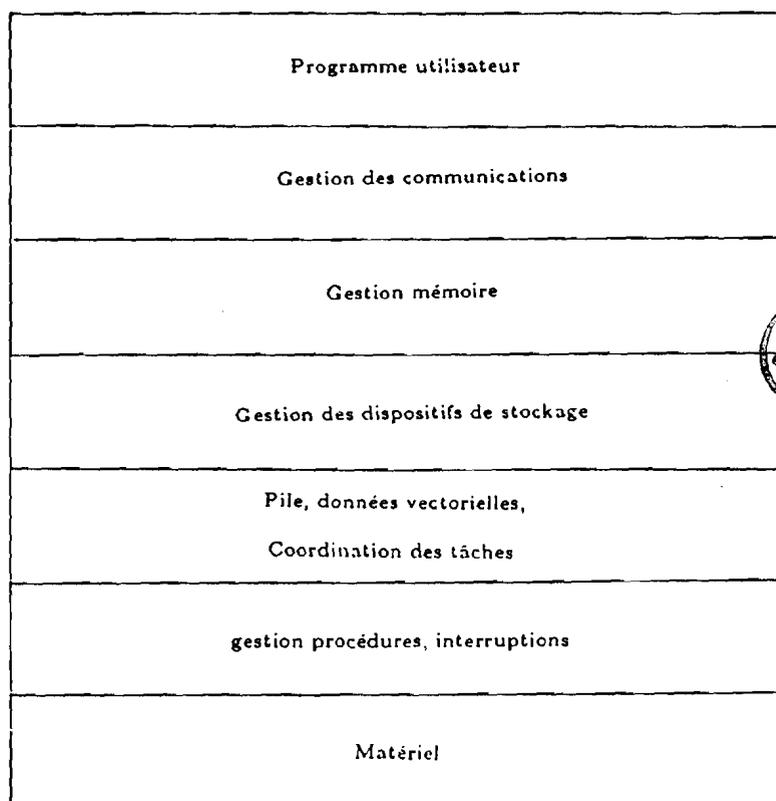


Figure 2-4

Le niveau trois supporte les mécanismes de coordination des tâches multiples. Tout processus peut être interrompu à tout moment. Il faut donc prévoir un mécanisme de synchronisation du déroulement des processus. Ce mécanisme comprend un mot d'état (sauvegarde des registres du processeur central) et dispositif de changement de contexte, qui attribue au processeur les nouvelles valeurs de registres pour la nouvelle tâche à effectuer ainsi que la restitution de l'état précédent. Ce dispositif n'est hélas pas suffisant, car le déroulement d'un processus est

généralement dépendant des résultats issus d'un, ou d'autres processus. C'est pourquoi, leur synchronisation s'avère nécessaire. Les sémaphores (cf paragraphe II-3-3-6) réalisent cette opération.

Le niveau quatre gère l'accès aux dispositifs de stockage du système. Ce niveau a besoin de sémaphores pour fonctionner. A chaque accès, le dispositif de gestion du stockage reçoit une demande de service, qui sera exécutée immédiatement ou mise en file d'attente.

Le niveau cinq gère l'allocation mémoire afin de pouvoir stocker tous les processus devant être exécutés. Le processus demandeur est en attente jusqu'à ce que les ressources nécessaires soient disponibles.

Le niveau six s'occupe des échanges entre processus. C'est un canal qui connecte deux processus entre eux. Le flot de données entre par l'une de ses extrémités et sort par l'autre. La gestion de la transmission est confiée à un pointeur d'écriture qui comptabilise le nombre d'octets reçus. Un canal sert également de moyen de communication entre ordinateurs. Ce système est utilisé pour relier l'ensemble des processus des ordinateurs en réseau. Ce niveau est également utilisé pour contrôler l'accès aux différents périphériques d'E/S tels que écran/clavier, imprimantes et autres.

Quant au niveau sept, il est constitué par un interpréteur de langage de commande évolué permettant à l'utilisateur de dialoguer avec le système.

Les paragraphes suivants décrivent les principes généraux utilisés pour la réalisation d'un système d'exploitation et qui sont mis en oeuvre lors du chapitre consacré à la réalisation.

### II-3-3) Principes de base

#### II-3-3-1) Introduction

La réalisation envisagée nécessite d'utiliser le principe mono-utilisateur, multi-tâches (multi-processeurs). Les difficultés de réalisation d'un tel système se résument ainsi: plusieurs programmes se trouvent présents dans un seul système informatique et se partagent les mêmes équipements. Il peut en résulter un risque dans la transmission des informations. Un moyen matériel de transmission ne doit véhiculer des informations qu'en provenance d'une seule source vers une seule destination. De même, certaines zones mémoire peuvent être adressées par plusieurs processeurs: il faut donc prévoir des moyens pour éviter qu'un programme ne vienne travailler dans une zone interdite, ou encore dans une zone commune autorisée, mais à un moment inopportun, afin d'éviter tout blocage du système. C'est ainsi que le système d'exploitation devra régir différentes tâches, de sorte qu'aucun processus ne puisse ni détruire un autre processus, ni monopoliser le processeur principal.

#### II-3-3-2) Noyau

Le noyau du système d'exploitation est composé de l'ensemble des routines qui prennent en charge les interruptions, assurent les synchronisations entre processus et constituent la structure élémentaire sur laquelle viennent se greffer les autres composants du système d'exploitation. Il est stocké en permanence dans la mémoire de l'ordinateur.

#### II-3-3-3) Processus

Un processus peut être assimilé à la notion de programme en exécution /KRAK/. Son évolution se déroule selon une séquence d'instructions qui compose le programme, modifie des données et accède aux ressources. A la différence des programmes qui sont des entités statiques décrivant uniquement les traitements

à effectuer, les processus constituent des entités dynamiques. C'est le processeur qui assure la transition d'un état au suivant en exécutant une instruction de programme. Un processeur ne peut exécuter qu'un processus à la fois, c'est pourquoi il faut faire la distinction entre les processus actifs et les processus passifs.

- Un processus actif (en cours d'exécution), est caractérisé par l'état du processeur et par l'état de la zone mémoire qui lui est affectée.

- Un processus passif n'est pas maître du processeur. Il est caractérisé par son état juste après l'exécution de la dernière instruction. Cet état est décrit par une variable (Etat\_Courant) écrite dans un formalisme voisin du Pascal.

```
type Registre = integer;
Etat_Processeur = record
    Regs : array [0..N] of Registre
    SP   : Registre;
    PC   : Registre;
    PSW  : Registre;
    BASE, TAILLE : Registre;
end;

var Etat_Courant : Etat_processeur;
```

Ou REGS représente les registres généraux du processeur, SP le pointeur de pile, PC le compteur ordinal, PSW les registres généraux, BASE un registre lié au mécanisme de gestion mémoire et TAILLE contenant la taille du programme.

Cette variable est sauvegardée dans une zone mémoire de façon à pouvoir être restituée au processeur lorsqu'il redeviendra actif. Cette zone mémoire est désignée sous le nom de bloc de contexte de processus. Cette variable est suffisante pour pouvoir reprendre le déroulement du processus mais insuffisante pour le système d'exploitation. En effet, rien ne lui indique

si tel ou tel processus est prêt pour reprendre son exécution. Le bloc de contexte du processus peut être parfaitement modélisé par la variable:

```

type Bloc_De_Contexte = record
    Suivant : ^Bloc de contexte;
    CM      : Etat-Processeur;
    Etat    : (Bloqué ou prêt);
    Pindic  : (Zone de paramètres);
end;

```

La variable 'Suivant' contient le processus qui vient ensuite dans la file d'attente des processus. Cette file d'attente est matérialisée par une liste chaînée (figure 2-5) des blocs de contexte décrite comme suit:

```

type File_Attente : record
    Queue, Tete : ^Bloc_De_Contexte;
end;

```

```

var File_Processus : File_Attente;

```

- Queue représente le dernier processus dans la file d'attente.

- Tête le premier processus dans la file d'attente.

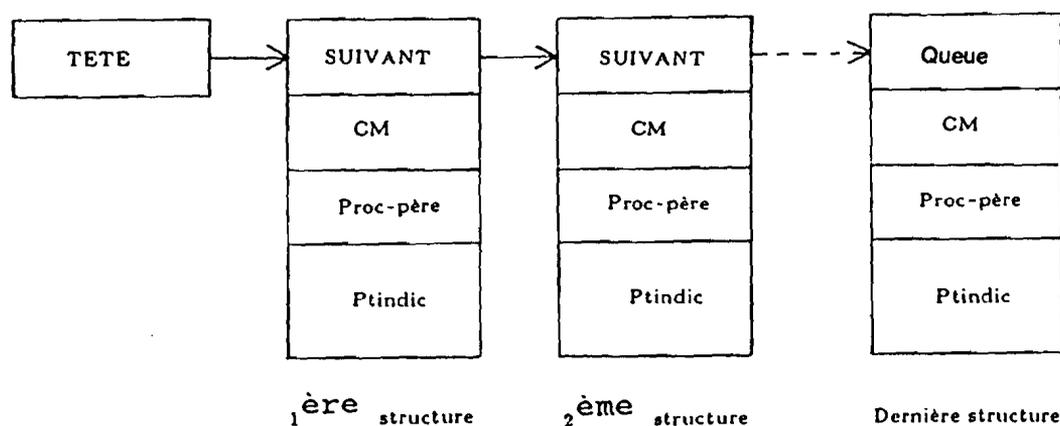


Figure 2-5

II-3-3-4) Gestion des ressources

Le système d'exploitation est responsable de la distribution des ressources non partageables entre les différents processus existant à un instant donné. Ces ressources matérielles (périphériques) ou logicielles (fichiers, données en mémoire) font l'objet de la part du processus qui en a besoin, d'une demande adressée au système d'exploitation. Lui seul connaît l'état de disponibilité de la ressource et peut arbitrer les demandes simultanées venant de divers processus.

Le fonctionnement général d'allocation d'une ressource est le suivant:

- Un processus adresse une demande de ressource au système d'exploitation.
- Si la ressource n'est pas disponible, le système d'exploitation place cette demande dans une file d'attente.
- Sinon, la ressource est attribuée et le processus demandeur poursuit son déroulement.

Ce mécanisme ne peut fonctionner sans demande de restitution de ressource:

- La ressource libérée est restituée dans l'ensemble des ressources disponibles.
- Le premier processus mis en attente pour disposer de cette ressource est activé.

Ces parties de programme introduisent la notion d'exclusion mutuelle, principe mis en oeuvre lorsque deux ou plusieurs processus cherchent à accéder simultanément à une même ressource qui ne peut être accessible que par un processus à la fois.

II-3-3-5) Exclusion mutuelle

Exemple d'application: considérons d'une part le cas d'un processus parcourant une chaîne de bloc de contexte et d'autre part un autre processus (déclenché par interruption pendant le travail du premier procesus) chargé de déplacer ce même bloc de contexte d'une file d'attente dans une autre. Lors de la reprise du premier processus, celui-ci travaillera avec la nouvelle file vers laquelle a migré le bloc de contexte. Il en résultera une exécution désastreuse du processus. Dans ce cas précis, la ressource doit être protégée contre tout accès intempestif jusqu'à ce que le processus en cours ait achevé son service. La chaîne de bloc de contexte nécessite un accès exclusif par un seul processus.

Pour assurer un accès exclusif à une ressource, il faut en fait assurer un accès exclusif au programme qui manipule cette ressource. Ce programme porte le nom de section critique. Pour ce faire, les processus demandeurs doivent respecter le protocole suivant:

- 1) Demander l'accès à la section critique,
- 2) Exécuter la section critique,
- 3) Libérer la section critique.

L'étape 1 est désignée par le nom WAIT (ou P) et permet de mémoriser une demande d'accès à une ressource.

L'étape 2 correspond au déroulement de la section critique.

L'étape 3 est désignée par le nom de SIGNAL (ou V) et libère la ressource en vue de permettre son utilisation par un autre processus.

Ce protocole est baptisé SEMAPHORE.

II-3-3-6) Sémaphore

Un sémaphore /CROC,DIJK/ est formé d'une variable entière et d'une file d'attente mémorisant les processus devant être servis. Il peut être décrit par la déclaration suivante:

```
type Semaphore = record
    E : integer;
    F : File_Attente;
end;
```

Cette variable est manipulée par trois primitives:

- Initialisation
- Demande de ressource
- Libération de ressource

Demande de ressource (WAIT)

```
procedure Wait (var S : Semaphore);
begin
    S.E := S.E-1;
    if S.E < 0 then Bloquer (S.F);
end;
```

La procédure Bloquer se déroule en quatre étapes:

- Bloquage du processus demandeur
- Mise en file d'attente
- Sélection du processus suivant à activer
- Récupération du contexte du processus réactivé

Par bloquer, on entend libération du processeur par le processus pour l'attribuer à un autre processus. On évite ainsi l'inactivité inutile du processeur.

Libération de ressource (SIGNAL)

```
procedure Signal (var S : Semaphore);  
begin  
  S.E := S.E+1;  
  if S.E < 0 then Activer (S.F);  
end;
```

Cette procédure libère le ressource d'un processus qui n'en a plus besoin pour l'attribuer, le cas échéant, à un des processus en attente dans la file.

Initialisation

```
procedure Init (var S : Semaphore; E0 : integer);  
begin  
  S.E := E0;  
  S.F.Tete := NIL;  
  S.F.Queue := NIL;  
end;
```

Cette procédure initialise la file d'attente à vide et fixe une valeur initiale à l'entier associé au sémaphore. Cette valeur correspond au nombre de processus pouvant exécuter la primitive Wait sans être bloqués. Dans le cas le plus courant, cette valeur est égale à un.

Les primitives P et V sont indivisibles, c'est à dire qu'un processus ayant commencé son exécution doit conserver le processeur sans le partager avec d'autres processus. Dans les systèmes monoprocesseur, cette clause est assurée en interdisant les interruptions pendant le déroulement de la primitive.

Exemple de fonctionnement

Soit trois processus P1, P2 et P3 cherchant à utiliser une ressource ne pouvant servir qu'un processus à la fois. Le sémaphore associé à la gestion de cette ressource enregistre les appels des processus

P1, P2 et P3 dans cet ordre. La figure 2-6 illustre les valeurs successives de la variable entière E et l'évolution de la pile F.

Evènement	E	File
Initialisation	1	
Wait par P1	0	
Wait par P2	-1	
Wait par P3	-2	
Signal par P1	-1	
Signal par P2	0	
Signal par P3	1	



Figure 2-6

II-3-3-7) Communication entre processus

Les processus se déroulant concurremment et nécessitant des besoins de communications pour transmettre des messages de toutes sortes sont gérés par un dispositifs de boîtes aux lettres.

Ce modèle de communication met en oeuvre un processus producteur qui dépose les messages dans la boîte aux lettres, et un processus consommateur qui les prélève lorsque le dépôt est achevé.

Ce transfert nécessite deux sémaphores (Arrivé et départ), correspondant respectivement au blocage du

producteur lorsque la boîte aux lettres est pleine, et du consommateur lorsque la boîte aux lettres est vide.

Initialement, la valeur de 'Arrivé' est égale à N (taille de la boîte aux lettres), et la valeur de 'Départ' est nulle. Ainsi, la valeur initiale de 'Arrivé' garantit que le producteur peut exécuter N fois de suite WAIT (Arrivé) sans être bloqué. Aussi, à chaque dépôt de message, le producteur active SIGNAL (Départ) pour que le consommateur puisse en disposer.

```
program Processus A;
begin
  :
  :
  Wait (Arrive);
  Deposer_Message;
  Signal (Depart);
  :
  :
end;
```

```
program Processus B;
begin
  :
  :
  Wait (Depart);
  Prendre_message;
  Signal (Arrive);
  :
  :
end;
```

On vérifie que ce processus respecte l'exclusion mutuelle: en effet, les valeurs initiales des deux sémaphores garantissent que le producteur passe avec succès WAIT (Arrivé), puis dépose le message, tandis que le consommateur reste bloqué sur WAIT (Départ) en attendant que le producteur ait passé SIGNAL (Départ).

### II-3-3-8) Gestion des échanges

Les échanges de données entre mémoire et mémoire de masse sont gérés par l'algorithme suivant:

```
begin
  Demande_Transfert;
  Transfert_Zone_memoire;
  Echange_Masse;
  Acquittement;
end;
```

- La première étape attend le dépôt d'une requête de transfert de données.
- La deuxième construit le programme qui localise le lieu du transfert.
- La troisième lance l'exécution du transfert.
- La quatrième s'assure que le transfert s'est déroulé correctement et bloque le processus s'il n'y a pas de dépôt de nouvelle requête.

### II-3-3-9) Gestion de la mémoire

Tout chargement d'un processus passe par une demande d'allocation mémoire. Après chargement initial, des demandes de mémoires supplémentaires peuvent être effectuées par les processus lors de leur exécution, et risquent ainsi de déborder dans la mémoire allouée à un autre processus, ou encore pourraient s'approprier toute la mémoire physique interdisant ainsi l'exécution d'autres processus. C'est pourquoi, il est nécessaire de prévoir l'occupation mémoire maximum d'un processus lors de son exécution.

Il existe également quelques cas où plusieurs processus peuvent être amenés à se partager une même zone mémoire. Le système de gestion de la mémoire doit donc contrôler l'accès à ces zones pour éviter le

blocage des processus. Le problème de l'interblocage est abordé de deux façons distinctes:

- La prévention, où l'algorithme garantit qu'une telle situation ne puisse se produire.
- La détection et guérison, où l'interblocage est détecté puis débloqué, ceci étant généralement accompagné d'une perte d'information.

L'unité d'allocation est définie par un découpage logique /LHER/ (segment ou page) afin de permettre au mécanisme d'implantation de charger un programme dans une zone mémoire quelconque. La politique d'allocation doit préciser où et quand charger un programme en mémoire. Les algorithmes les plus courants utilisent le chaînage. Chaque zone libre est associée à un descripteur qui contient son adresse d'origine, sa taille et des liens de chaînage qui indiquent l'adresse de l'emplacement libre suivant. Les tests d'allocation consistent à explorer les zones libres pour trouver l'emplacement de taille suffisante qui peut recevoir le programme à charger. Dans le cas où les programmes ou données utiles ont une taille fixe prédéfinie, l'algorithme d'allocation est considérablement simplifié.

#### II-3-3-10) Langage de commande

Le langage de commande permet à l'utilisateur d'ordonner à la machine d'effectuer les fonctions qu'il souhaite. Les commandes de l'utilisateur sont traitées par un programme que l'on nomme "analyseur de commande". Ce programme est le point d'entrée du système d'exploitation. A chaque fin d'exécution d'une tâche dictée par l'utilisateur, le système revient sur ce point d'entrée. Le schéma général de l'interpréteur de commande est donc équivalent à une boucle sans fin:

```
begin
  Initialisation;
  Traitement_Commande;
  if Ok then
    Execution_Commande;
  else
    Traitement_Erreur;
  endif
end;
```

La séquence initialisation consiste à vider le buffer de réception des commandes, fermer tous les fichiers mis en jeu lors de la commande précédente et se mettre en attente d'une nouvelle commande.

Lorsque ce processus détecte une fin de commande, la séquence Traitement\_commande en entreprend l'analyse et lance l'exécution si la commande fait partie d'une liste standard au système. Sinon, un traitement d'erreur est effectué, affiche son diagnostic et se met en attente d'une nouvelle commande.

#### II-3-3-11) Conclusion

Cette description des mécanismes des systèmes d'exploitation permet de remplir toutes les fonctions souhaitées par les utilisateurs. Ces techniques les plus couramment rencontrées résolvent les problèmes de conceptions et constituent une base efficace pour l'implantation d'un système sur une machine mono-utilisateur et multi-processeurs.

#### II-4) Système de développement

Le système de développement est l'outil idéal pour mener à bien l'écriture et la mise au point des programmes orchestrant l'ensemble de la réalisation. Un système de développement est lié à une famille de microprocesseur. La base de notre système étant architecturé autour du microprocesseur 8085 d'INTEL, l'outil de développement utilisé est donc l'INTELLEC 261. Le développement logiciel s'effectue à l'aide d'un

éditeur de texte quelconque permettant l'écriture des programmes. Le développement matériel et la mise au point sont réalisés par un émulateur, sorte de sonde venant s'enficher à la place du microprocesseur pour en simuler parfaitement le fonctionnement et permettre à l'utilisateur d'avoir accès à toutes les données transitant par celui-ci.

#### II-5) Phases de développement

La figure 2-7 résume les différentes étapes nécessaires à la mise au point d'une application à microprocesseur.

Les étapes 1 et 2 représentent l'aide à l'écriture et à la transcription du programme en langage compréhensible par le microprocesseur.

L'étape 3 constitue l'émulation en temps réel du programme sur le matériel, et l'implantation du logiciel en état de marche sur des mémoires non volatiles.

La dernière étape consiste à libérer l'application de toutes ces "attaches" nécessaires au développement et bien étendu à vérifier qu'elle fonctionne.

#### II-6) Conclusion

Ce chapitre a été consacré à la présentation de tous les outils, tant matériels que logiciels utiles pour la réalisation du projet. Au stade de cette étude, il est maintenant nécessaire d'exposer les méthodes permettant le dialogue homme/machine afin d'exploiter au mieux les résultats produits par le système.

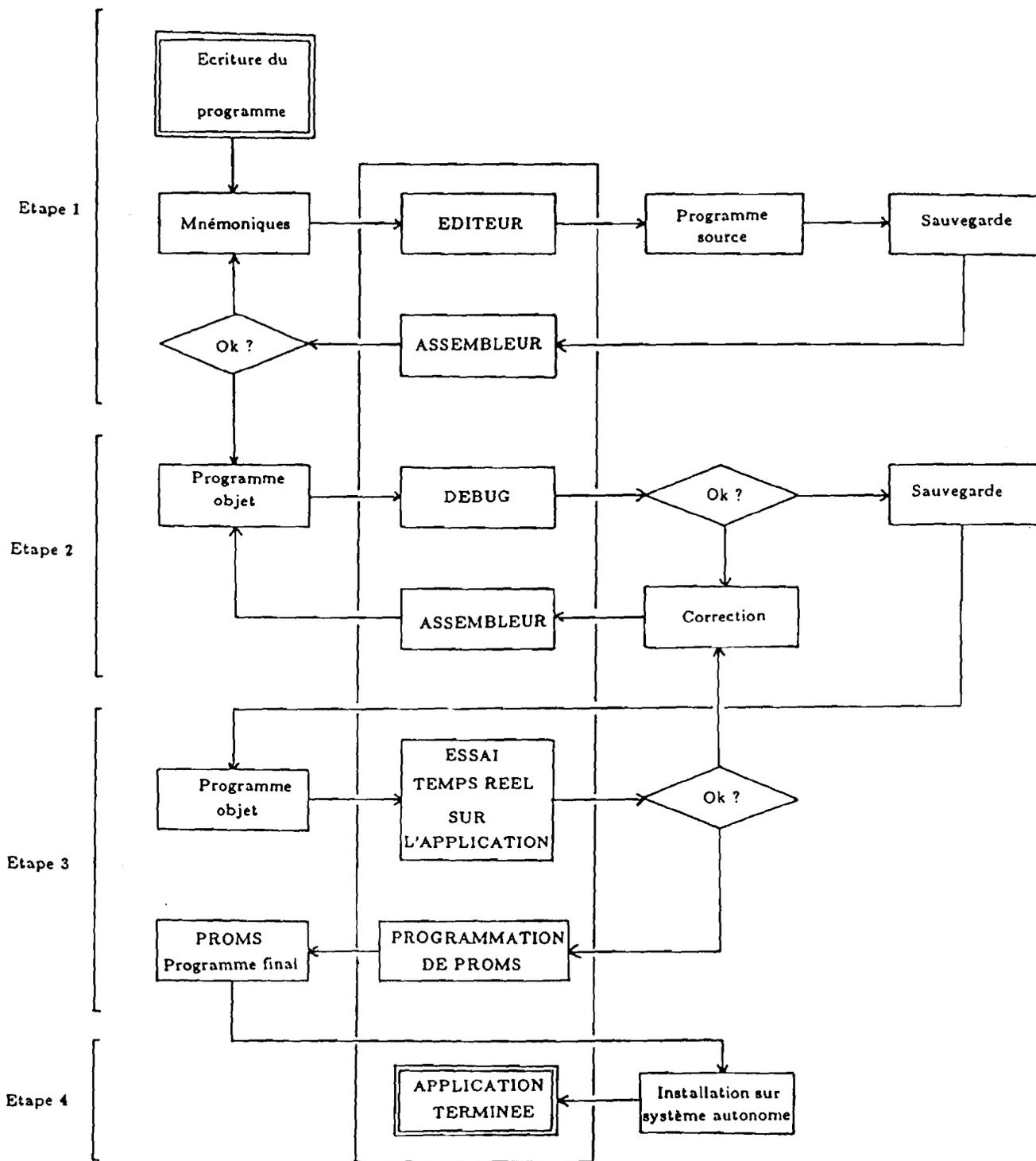


Figure 2-7

III) DIALOGUE HOMME MACHINEIII-1) Introduction

"Un système de traitement du signal se comporte comme un processus en constante évolution. Ce processus est soumis à des informations extérieures incontrôlables mais mesurables (données) auxquelles il réagit selon des procédures parfaitement définies (logiciel). La sortie du dispositif est l'élément permettant à l'opérateur d'apprécier à chaque instant l'état d'évolution du système de traitement et de moduler ce traitement, donc les résultats, pour des modifications ou pour l'introduction de nouvelles informations dans le processus."/VASS3/

Le principal élément de communication entre un processus et l'opérateur réside dans la représentation des résultats affichés sur des écrans de visualisation ou une imprimante.

Les résultats représentés sous forme de tableaux de chiffres ne fournissent pas une capacité d'interprétation immédiate. Une représentation graphique des résultats a le mérite de synthétiser les résultats au plus haut niveau /THOR/.

III-2) Présentation des résultatsIII-2-1) Imagerie

Les résultats issus d'une analyse quelconque peuvent être représentés sous diverses formes. Sur la figure 3-1 la synthèse de trois paramètres est effectuée par un triangle équilatéral /BERT/. Chaque côté du triangle représente l'un des paramètres. Les valeurs caractéristiques des paramètres sont reliées entre elles par des droites afin de dessiner un triangle inscrit dans ce triangle équilatéral. Ainsi en fonction des différentes valeurs des paramètres, on obtient diverses formes de triangles, chacune étant représentative d'un état déterminé. Certaines formes

les plus couramment rencontrées peuvent aussi être distinguées en effectuant un changement de couleur du triangle.

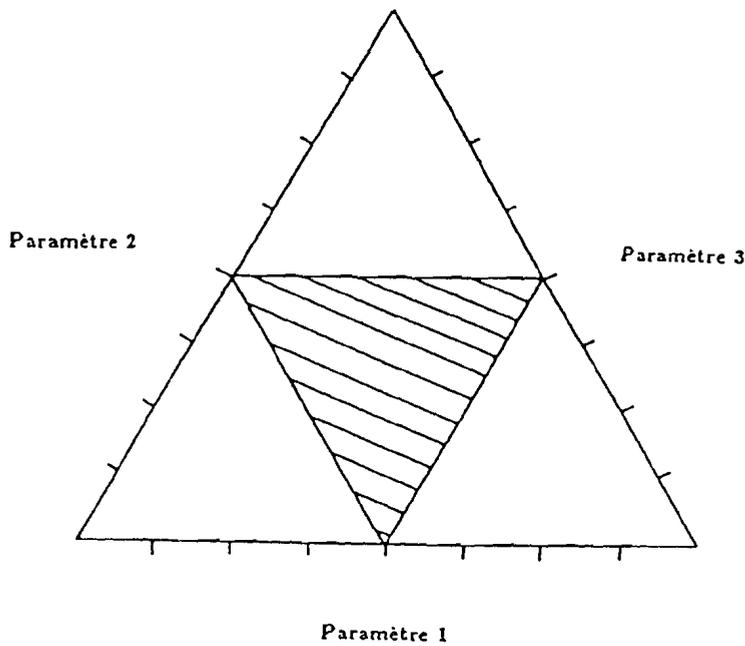


Figure 3-1

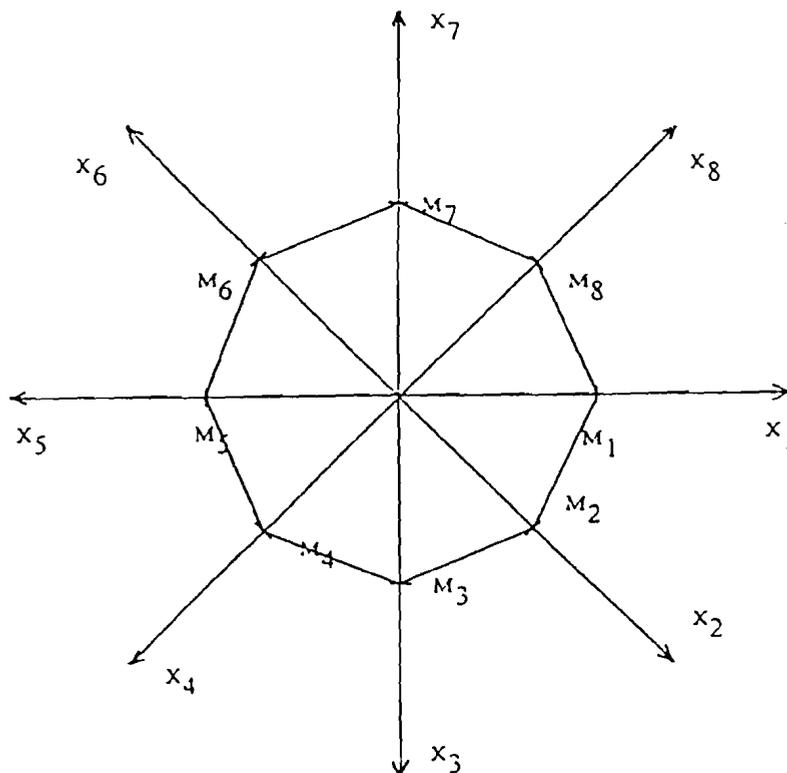


Figure 3-2

L'exemple de la figure 3-2 met en évidence huit variables d'un bilan quelconque auxquelles on applique

la règle suivante dans le but d'améliorer et de clarifier la représentation:

-Sur chaque axe  $x_i$ , l'origine des graduations est choisie de telle sorte que les moyennes  $m_i$  (ou valeurs normales) de chacune des variables  $x_i$  soient distribuées sur un cercle de centre O /WILL/.

Si tous les résultats appartiennent à l'échelle normale, un polygone quelque peu régulier ayant un nombre de côtés correspondant au nombre de branches se développe lorsqu'on relie les points des résultats correspondant à chaque branche. Les échelles étant dans une position constante, l'utilisateur peut rapidement se familiariser avec les formes et les modèles ayant une signification particulière. Par exemple, quand les échelles sont mises en place de façon appropriée, la distorsion du polygone apparaissant nord-ouest peut refléter un certain type de pathologie, orienté nord-est un autre type, etc ...

Il est donc important de bien disposer ces échelles pour produire des graphes typiques de maladies.

Dans ce format de représentation, le temps peut être représenté comme étant l'une des branches du graphique ou sous forme d'une animation dans laquelle chaque modèle est successivement remplacé par celui qui suit, l'intervalle de temps entre les différents tracés étant relié à l'intervalle entre les observations.

Cette représentation a été retenue car elle présente de nombreux avantages:

- Elle effectue la synthèse d'un nombre important de paramètres: jusqu'à huit pour être encore aisément interprétable contre trois pour la représentation triangle. Il est néanmoins possible de faire figurer un nombre plus important de paramètres en utilisant deux ou trois triangles, mais l'interprétation des figures est plus problématique.

- Les polygones possèdent une déformabilité importante et peuvent représenter un nombre élevé de pathologies.
- L'apprentissage d'un ensemble de figures est relativement facile. Un diagnostic peut donc être formulé rapidement.

### III-2-2) Variables visuelles

La construction d'une représentation graphique consiste à transcrire chaque composante de l'information par une variable visuelle afin d'en faciliter la compréhension.

Une bonne transcription graphique nécessite six variables visuelles /BERI/:

- Les deux variables du plan (X,Y)
- La taille
- La couleur
- L'orientation
- La forme

De plus, la règle suivante est appliquée: un symbole et sa légende doivent être de la même couleur.

### III-2-3) Histogramme

L'histogramme est une représentation graphique traditionnelle. C'est un instrument efficace d'édition graphique des résultats. Les barres représentent sous forme statistique les informations recueillies après traitement.

"Cette représentation offre deux possibilités:

a) Elle permet de faire ressortir la fréquence d'apparition d'un évènement à l'intérieur de diverses catégories d'une composante ordonnée comme par exemple la fréquence cardiaque et les rythmes respiratoires (figure 3-3).

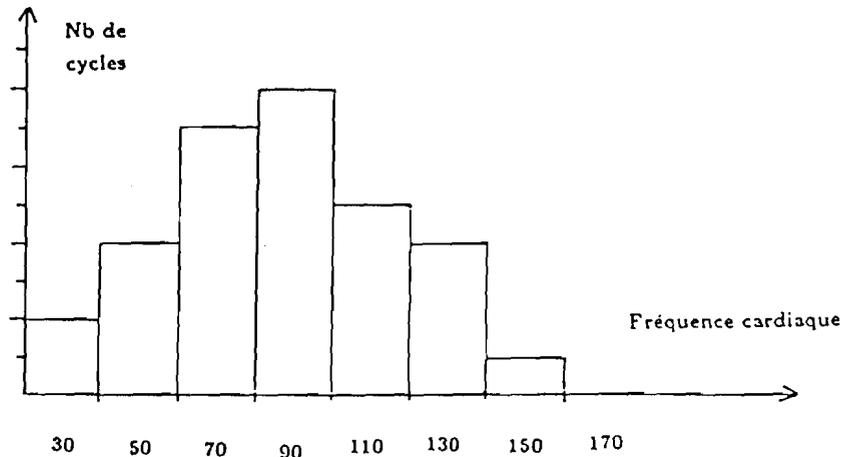


Figure 3-3

b) Elle permet de représenter une information relative à une composante non ordonnée (figure 3-4). Dans ce cas, on peut classer les composantes de l'histogramme de manière à présenter une image plus facilement mémorisable étant donné que les barres (pathologies ou autres) sont classées dans un ordre croissant." /VASS3/

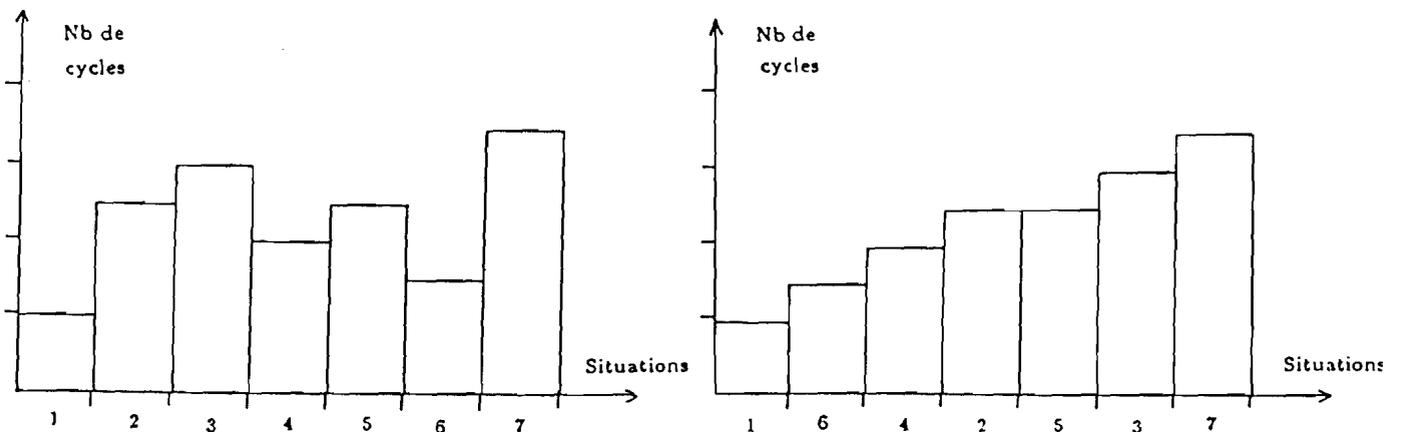


Figure 3-4

III-3) Conclusion

L'aspect critique de la représentation graphique consiste à éviter une représentation trop chargée, aux informations redondantes, qui présentent un risque de

fatigue visuelle lors de la surveillance. Il est donc nécessaire d'appliquer une succession de règles ergonomiques portant sur les couleurs, les formes, les caractères alphanumériques utilisés et la densité d'informations.

### III-5) Dialogue Homme-Machine

#### III-5-1) Introduction

Une instrumentation micro-informatisée suppose la présence d'une console de dialogue qui facilite l'utilisation du système par un opérateur possédant ou non des connaissances en informatique.

#### III-5-2) Interactivité

Un système interactif d'aide à la décision permet d'afficher des données sous diverses formes (tableaux, courbes ...), de combiner ces données et de faire appel à des programmes statistiques.

Les consoles de dialogue sont généralement constituées d'un ensemble de touches de fonctions afin de faciliter l'utilisation par l'opérateur, chaque touche correspondant à un fonctionnement prédéfini de l'appareil.

Les consoles de dialogue spécialisées sont constituées d'un écran et clavier standard, clavier généralement composé de trois zones:

- Une zone de touches alphanumériques générant le code ASCII

- Un pavé numérique

- Des touches de commandes jouant le rôle de touches de fonctions.

Ce type de console est totalement dédié à l'interactivité et à l'aide à la décision. Son

utilisation est certainement beaucoup plus lourde pour un non informaticien, aussi, l'implantation d'un logiciel gérant toutes les erreurs syntaxiques s'avère indispensable pour guider l'opérateur dans l'utilisation du système.

### III-5-3) Contraintes logicielles

La construction et l'exploitation des informations numériques médicales sont complexes. Un logiciel adapté doit être capable de réaliser un modèle de calcul utilisable avec toutes les options envisagées, et de fournir en quelques secondes un état précis et complet d'une situation. De même, la représentation graphique des données numériques est délicate car elle nécessite un nombre relativement important de calculs et de choix d'échelles.

"L'information représentée par une suite de variables numériques n'a pas seulement une valeur de document, mais c'est aussi le moyen de représenter l'évolution d'une situation. Les informations liées à un système de représentation graphique fournissent un moyen d'aide à la décision très appréciable. C'est l'accès à ces informations qui prédomine la convivialité d'un système.

La gestion d'une console de dialogue comporte un logiciel important chargé en grande partie de gérer de nombreuses manipulations de données issues du dialogue opérateur-système. Quelques méthodes simples permettent d'établir un dialogue entre la machine et l'opérateur.

La méthode des questions réponses (par oui ou non) permet une réalisation logicielle simple et, pour l'utilisateur une progression rapide vers une tâche de traitement. Cette solution n'est pas conforme pour une machine qui prétend présenter des résultats graphiques d'une façon claire et attrayante.

Une gestion par menu permet à l'utilisateur de se diriger plus rapidement vers l'option choisie. Cette

gestion peut être agrémentée par plusieurs fonctions permettant de revenir sur certaines décisions ou encore sauter certaines étapes. Néanmoins, il faut prendre garde à ne pas trop surcharger l'emploi de la console. De plus, il est nécessaire de lever toutes les ambiguïtés sur l'utilisation de l'appareil. Les menus doivent être parlants afin de ne pas laisser l'opérateur dans l'expectative."/VASS3/

#### III-5-4) Conclusion

Quel que soit le degré de sophistication du matériel sur lequel est construite la machine, c'est grâce au logiciel que l'utilisateur a ou non l'impression d'avoir affaire à un système amical, c'est à dire lui fournissant une aide réelle et ne l'obligeant pas à tenir compte de particularités de type informatique sans rapport avec son travail. L'interface homme/machine doit donc être particulièrement soignée pour que l'utilisation du système soit facile et pour être adaptée aux contraintes imposées par l'environnement extérieur. La réalisation de l'interfaçage homme/machine est très importante car elle constitue l'unique lien entre les applications et le monde extérieur. Si l'interface n'est pas satisfaisante (utilisation trop complexe, temps de réponse demande/résultat élevé), la convivialité du système s'en trouvera affectée ainsi que son utilisation.

IV) REALISATION PRATIQUEIV-1) Introduction

On distingue deux types de modules dans le système: les cartes maîtres et les cartes esclaves. Les modules maîtres sont des cartes à microprocesseur qui constituent des modules "intelligents" et les modules esclaves sont des ressources communes accessibles par tous les modules maîtres du système. Tous ces modules sont reliés entre eux par l'intermédiaire du multibus.

IV-2) Synoptique général

L'architecture générale du système est décrite figure 4-1 /DOMI/. Les modules maîtres contiennent les routines système et le logiciel d'application. Les ressources communes sont constituées par la mémoire principale, la mémoire de masse et les interfaces d'Entrées/Sorties. Quant au terminal, il ne dépend que du module maître Ms qui joue le rôle d'interface homme-machine et est chargé de la gestion globale du système.

Chaque module maître peut avoir accès par l'intermédiaire du multibus aux ressources communes et peut le contrôler par l'intermédiaire des circuits logiques lui permettant l'accès ou non. On distingue donc les modules actifs et passifs. Tous les modules maîtres sont actifs car ils commandent l'accès aux ressources communes via le multibus. Les modules esclaves sont généralement passifs. Certains peuvent prendre le contrôle du multibus mais avec l'accord d'un module maître.

Suivant l'architecture ainsi définie, sur la figure 4-1 apparaissent:

- Les modules Mi (Processeurs d'entrée) qui effectuent le traitement local des signaux.
- Le module de synthèse Ms (Processeur maître) qui gère l'ensemble du système. Il collecte tous les résultats fournis par les modules Mi, les

synthétise selon les ordres reçus de l'extérieur et contrôle les échanges d'informations effectués à travers le multibus.

- Le module B.A.L. (Boîte aux lettres) est une mémoire tampon chargée de stocker provisoirement les résultats issus des modules  $M_i$ , et les variables de gestion de l'ensemble du système. Cette mémoire est accessible par les modules  $M_i$  et  $M_s$ .

- La mémoire de masse  $M_m$  stocke tous les résultats des modules  $M_i$  par tranche d'un quart d'heure.

- La carte C.A.D. (Convertisseur analogique digital) fournit aux modules  $M_i$  les échantillons numériques des signaux.

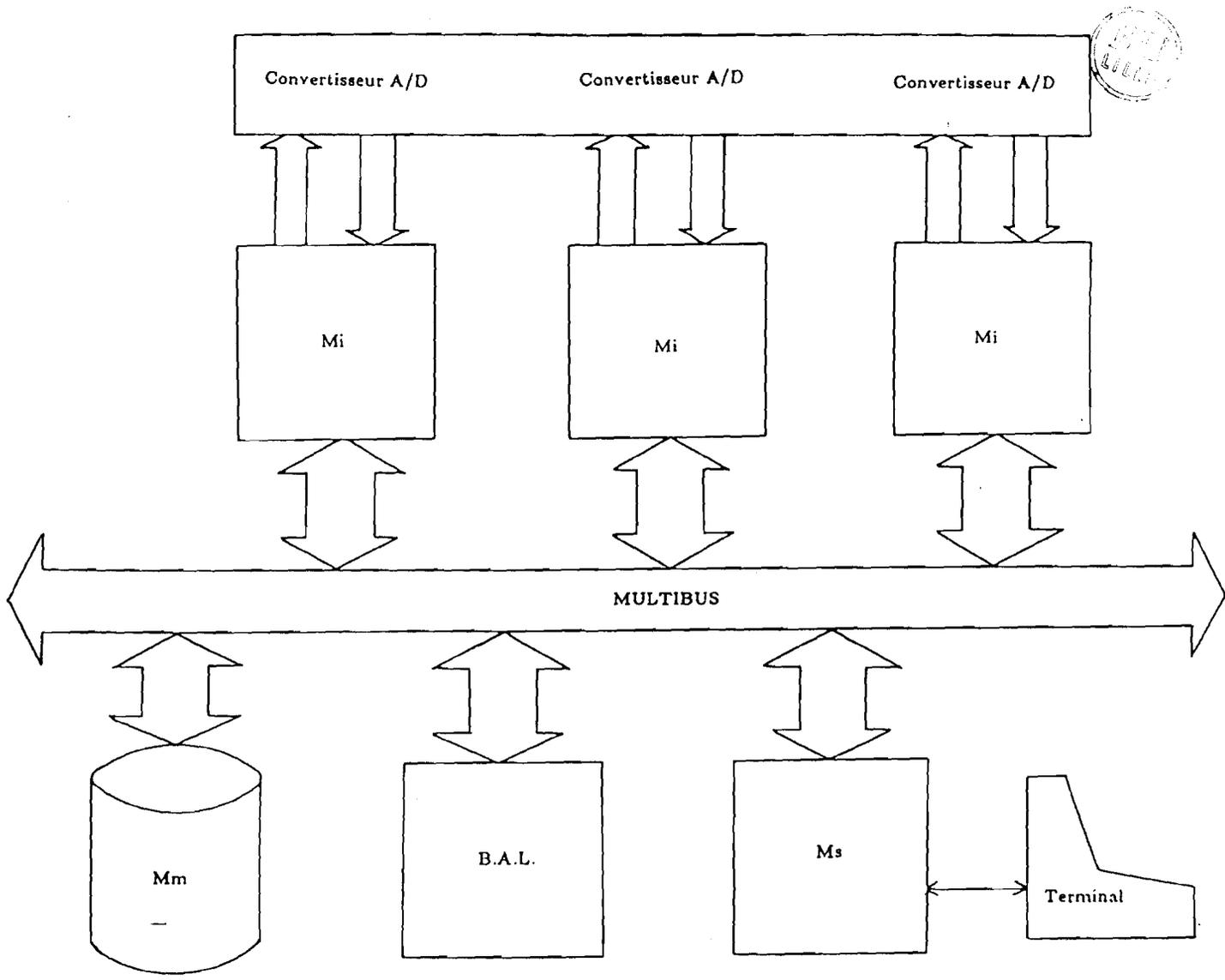


Figure 4-1

IV-3) Carte C.A.D.

Les modules Mi reçoivent leurs échantillons par la carte de conversion Analogique/Digitale. Cette carte supporte trois convertisseurs 8 bits et les éléments nécessaires pour l'amplification et la mise en forme des signaux. Le convertisseur alimentant le module chargé du traitement des signaux respiratoires (carte PVC) fonctionne avec un multiplexeur pour faire l'acquisition des trois signaux respiratoires.

IV-4) Carte B.A.L

Le module B.A.L. /INT6/ est une mémoire tampon chargée de stocker provisoirement les résultats issus des modules Mi et les variables de gestion de l'ensemble du système.

Cette carte a été structurée en fonction de l'occupation mémoire nécessaire aux modules Mi. Elle est découpée en trois zones, chacune contenant les indicateurs de travail et l'emplacement réservé au transfert des résultats.

IV-5) Carte Mm

La mémoire de masse Mm /INT3/ stocke tous les résultats des traitements des modules Mi à concurrence de vingt-quatre heures. Cette mémoire d'une capacité de 128 Koctets est divisée en 2048 pages de 64 octets. Chaque page peut contenir le résultat du traitement d'un quart d'heure d'une situation, ou la marque d'une situation toutes les quinze secondes sur un quart d'heure.

IV-6) Modules Mi

Chaque module Mi /INT2/ est chargé du traitement local d'une fonction physiologique abordée par le

ystème. Ces modules possèdent la structure suivante (figure 4-2):

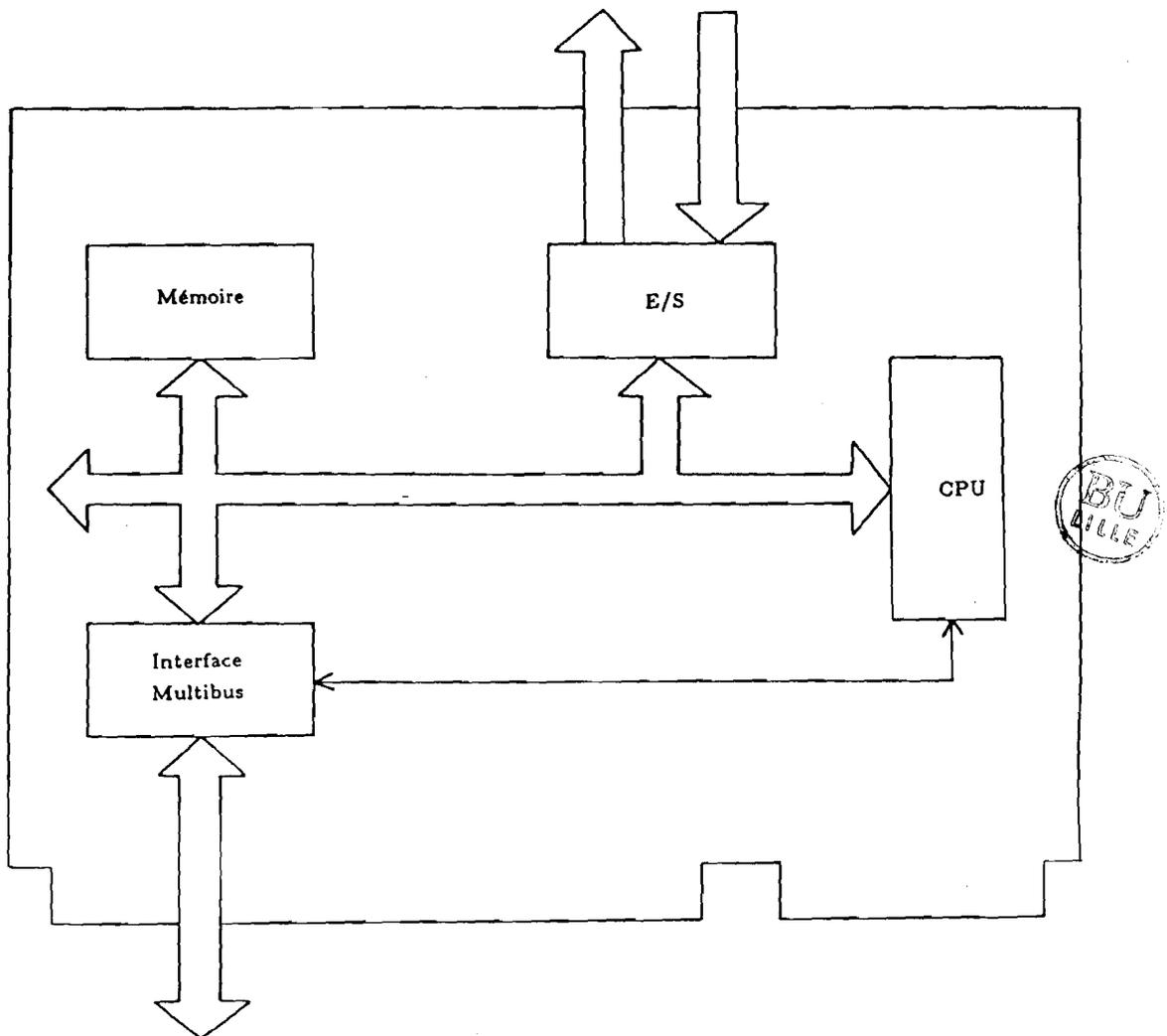


Figure 4-2

L'ensemble de traitement conventionnel est composé des éléments suivants:

- microprocesseur
- mémoire morte et vive
- interface d'entrée/sortie parallèle

Les entrées/sorties parallèles commandent la réception des échantillons. Pour une fréquence d'échantillonnage de 50 Hz et un temps moyen d'exécution d'une instruction de 5  $\mu$ s, le microprocesseur peut exécuter 4000 instructions entre l'acquisition de deux échantillons ce qui est largement

suffisant pour calculer les deux informations suivantes:

- Un vecteur de paramètre caractérisant l'évènement (pour l'E.C.G., largeur QRS et fréquence cardiaque)
- Le résultat d'une classification affectant l'évènement à une situation (arythmie à complexe normal, fin ou large)

Chaque module Mi réalise donc de manière autonome et en temps réel:

- L'acquisition du signal
- La génération des vecteurs de paramètres caractérisant chaque évènement
- La classification des évènements
- L'élaboration et la mémorisation du passé récent (quart d'heure) sous forme d'histogrammes de répartition des paramètres pour chaque situation.

#### IV-7) Module Ms

##### IV-7-1) Description

Le principal module maître /INT1/ du système est doté de tous les éléments nécessaires au fonctionnement classique d'un micro-ordinateur (figure 4-3).

- Microprocesseur.
- Mémoire vive, mémoire morte.
- Entrées/Sorties parallèle et série.
- Contrôleur d'interruption.

Le bus interne n'est pas accessible de l'extérieur. Il est utilisé par le microprocesseur hôte pour communiquer avec ses propres ressources. La mémoire et les E/S propres de la carte sont donc garanties contre tout accès extérieur. De plus, le microprocesseur local est ainsi protégé contre les pannes possibles des autres microprocesseurs externes.

Parmi les ressources internes de cette carte, on distingue:

- Capacité d'adressage de 32 K d'EPRAM et de 4 K de RAM.
- Un contrôleur d'interruption 8259A.
- Deux circuits d'E/S parallèles 8255.
- Un circuit d'E/S série 8251A.
- Un triple compteur 8253A.

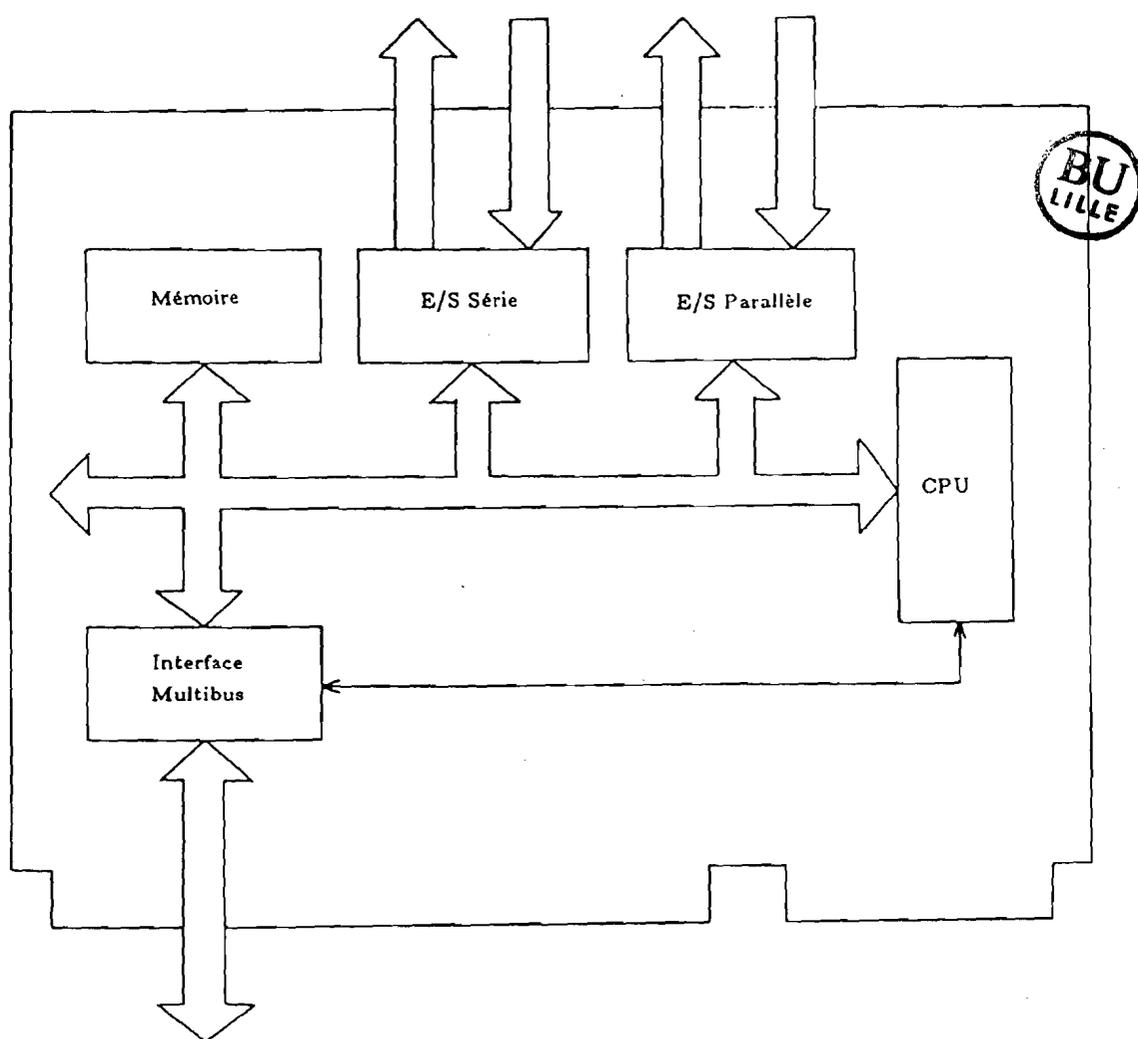


Figure 4-3

Toutes ces ressources sont assignées à des adresses précises qui peuvent être modifiées par straps (configurations ROM) ou par d'autres PROM qui sont utilisées pour décoder principalement les E/S.

La sélection du bus interne ou externe dépend de l'assignation des adresses générées par le microprocesseur. Si aucune de ces adresses n'est adressable sur la carte, alors le contrôleur de bus est activé et permet ainsi l'accès à la ressource commune sélectionnée.

Le contrôleur d'interruption est capable de gérer huit niveaux d'interruption. Les interruptions peuvent provenir de circuits présents sur la carte ou du multibus.

Le microprocesseur peut lui-même recevoir quatre niveaux d'interruption dont une non masquable qui peut être utilisée en cas de coupure secteur pour sauvegarder l'état du système.

#### IV-7-2) Rôle

Le module de synthèse Ms supporte le système d'exploitation de l'ensemble. Il assure la coordination des modules Mi par l'intermédiaire de la mémoire B.A.L., la gestion de la mémoire de masse et du terminal. Il collecte l'ensemble des tranches élémentaires d'un quart d'heure fournies par les modules Mi et les synthétise sur une période de temps selon les commandes reçues de l'extérieur. Il gère l'ensemble des indicateurs utilisés pour la synchronisation avec les modules Mi. Ces indicateurs sont positionnés, soit par l'intermédiaire de l'opérateur pour avoir accès aux résultats des examens physiologiques, soit par le système d'exploitation tous les quarts d'heure pour permettre un stockage des résultats dans la mémoire de masse. Le système d'exploitation assure également la synchronisation des sémaphores chargés de la gestion des deux ressources allouées au module Ms, le microprocesseur et la console de dialogue.

Le module Ms gère également les alarmes. L'alarme est le résultat de la scrutation de l'ensemble des

modules Mi. Toutes les alarmes peuvent être déclenchées, soit par un signal seul, soit par un enchaînement d'anomalies détectées sur les signaux évoluant simultanément.

#### IV-7-3) Fonctionnement global

Un compteur interne est la base du fonctionnement de ce module. Il permet:

- De tester les situations toutes les secondes et de générer des alarmes s'il y a lieu.
- De positionner les indicateurs de transfert tous les quarts d'heure, autorisant ainsi les modules Mi à transférer leurs résultats dans la carte B.A.L.
- De gérer une horloge temps réel qui permet l'affichage des Heures-Minutes-Secondes réelles ainsi que le temps écoulé depuis l'initialisation de l'appareil.
- De plus, il entretient la synchronisation temporelle afin d'ordonner les événements, et constitue un moyen de mesure absolue utilisé par un échéancier chargé d'exécuter divers processus en fonction d'un horaire fixé.

Le fonctionnement détaillé de ce module est repris au chapitre V.

#### IV-8) Console de dialogue

Le clavier à la disposition du personnel soignant permet le dialogue avec l'appareil. Ce dialogue est réalisé sous forme de menu afin de permettre un accès facile aux différentes fonctions offertes à l'opérateur telle que:

- Visualisation d'un bilan pour un signal dans une tranche de temps donnée (présent ou passé).

- Mise en évidence des relations existant entre plusieurs signaux.
- Visualisation des tendances d'évolution d'un signal donné vers tel ou tel état.
- Visualisation du signal par affichage de la valeur des échantillons.

La console Lyme 6000 est reliée au module Ms par une liaison série fonctionnant à 9600 bauds. Ce terminal possède huit pages alphanumériques 80 ou 123 colonnes et 2 pages graphiques 300x400 points, mixable avec n'importe quelle page alphanumérique, le tout en 8 couleurs. Le clavier est du type qwerty complété par un pavé numérique et de 11 touches de fonctions. Toutes ces possibilités sont amplement exploitées et accessibles à l'utilisateur par un dialogue aisé et facilement utilisable. La représentation des résultats fait largement appel aux techniques de l'imagerie.

L'utilisation et le fonctionnement de la console est décrit au chapitre V.

#### IV-9) Conclusion

Les matériels qui viennent d'être détaillés constituent un ensemble inerte. Pour pouvoir les animer, il faut maintenant étudier les logiciels qui vont établir les relations entre les matériels d'une part, l'opérateur et ces mêmes matériels d'autre part.

V) FONCTIONNEMENT DE L'APPAREILV-1) Console de pilotageV-1-1) Introduction

Ce système de surveillance doit non seulement permettre le monitoring /HEW1, HEW3/ mais également l'établissement d'un bilan physiologique et l'aide à l'examen clinique /VAS1/. Il est donc nécessaire de donner une grande importance à l'interactivité et à la visualisation des résultats au niveau de la console. La préférence a été donnée à la conception de représentations graphiques, plus facilement interprétables qu'une longue liste de nombres. De même, une mise en page faisant appel à des artifices divers de représentation permet de faire ressortir les résultats aux dépens de points moins importants de l'écran.

V-1-2) Représentation des résultats

Mettre des résultats en évidence consiste à les particulariser en utilisant des artifices habituels tels que: souligner, encadrer, décaler, décider leur impression en des points forts d'une page.

Disposer consiste à fixer la position relative des résultats les uns par rapport aux autres et à imposer leur coexistence sur une même page.

L'élaboration de toute mise en page se caractérise par deux étapes:

- L'étape "composition" qui a pour but de mettre en évidence des informations ou fixer leur position relative les unes par rapport aux autres, donc à fixer la position du plus petit rectangle dans lequel ces informations peuvent s'inscrire.

- L'étape "mise en page" qui répartit les rectangles informations en imposant leur

implantation en des points forts du support fictif.

Ainsi, en respectant ces règles, le logiciel est composé de déclarations qui permettent la description et la génération d'informations qui lui sont propres, et d'instructions pour utiliser à des fins précises les informations qui leur sont communiquées.

Pour faciliter la lecture et l'écriture sur l'écran, la démarche générale de constitution d'une fenêtre a été adoptée.

La création d'un cache donne la forme générale de l'ensemble du texte. Le choix d'un emplacement particulier et la taille d'un mot-clé permet de souligner un mot pour attirer l'attention de l'utilisateur. Le texte de base est introduit automatiquement au bon endroit. La modification du texte (insertion, suppression...) est facilitée par l'utilisation d'un éditeur type "pleine page".

Ces différentes règles ont été utilisées pour concevoir un dialogue opérateur/système présenté ci-après.

### V-1-3) Fonctionnement

C'est par l'intermédiaire de la console LYME 6000 que l'opérateur initialise l'appareil et peut lui demander les renseignements concernant l'état physiologique du patient.

A la mise sous tension, la première étape remplie par l'appareil consiste à afficher la configuration par défaut et à demander si l'on souhaite la modifier (figure 5-1-1). En cas de réponse positive, un curseur se positionne en fin de première ligne au niveau du premier masque de saisie pour permettre à l'opérateur de modifier les paramètres du système. Un éditeur permet de se positionner au niveau de toutes les réponses à

introduire en utilisant les touches de déplacement du curseur.

Etalonnage capteurs	
CO <sub>2</sub>	: 2,5 V = 5%
Volume	: 2,5 V = 1 L
Pression	: 2,5 V = 50 Cm H <sub>2</sub> O
Caractéristique appareil	
Type d'appareil: 1)RCD 2)UV1 3)H.P. 4)Autre	
VDapp (ml)	:
Capp (ml/cmH <sub>2</sub> O):	:



Figure 5-1-1

Le curseur peut tout d'abord se déplacer dans les quatre directions:

- Touche ↑ : Déplacement vers la ligne précédente
- Touche ↓ : Déplacement vers la ligne suivante
- Touche ← : Déplacement vers le caractère de gauche
- Touche → : Déplacement vers le caractère de droite.

Le curseur est limité dans ses déplacements par les masques de saisie dévolus à l'entrée des paramètres.

L'utilisateur dispose de deux modes de saisie:

- Le mode "remplacement": le caractère frappé remplace celui placé sous le curseur qui est ensuite décalé d'une position vers la droite.
- Le mode "insertion": la frappe d'un caractère entraîne un déplacement de la portion de ligne

allant du curseur à la fin de la ligne d'une position vers la droite. Le nouveau caractère prend place dans l'espace ainsi libéré.

Deux autres touches sont une aide précieuse à l'édition:

- La touche < recule le curseur d'une position et détruit le caractère placé sous le curseur.

- La touche "CR" actionnée lorsque le curseur est en fin de ligne permet de passer à la ligne suivante pour poursuivre l'introduction de nouveaux paramètres.

Lorsque le curseur est positionné sur la dernière ligne, la frappe de la touche "CR" valide la configuration et permet le passage à la première page du système.

Lorsqu'une reconfiguration s'avère inutile, le système affiche également la page 1.

#### V-1-4) Présentation de la page 1

Cette étape constitue la première action à effectuer par le personnel soignant. Elle a pour rôle de définir un fichier se rapportant au patient.

La page 1 comporte un cadre de dix lignes divisé en deux parties. A gauche une marge où est inscrit un questionnaire, et à droite l'emplacement réservé pour les réponses. Dans ce qui suit, on considère uniquement la partie droite, qui seule est accessible par l'opérateur pour lui permettre de rentrer un minimum d'informations: le nom du patient et l'heure (figure 5-1-2).

Nom	
Date	
Heure	
Surf. corporelle (m <sup>2</sup> )	
Temp. (Degré c )	
Remarques	

Figure 5-1-2

A l'origine, le curseur est situé en haut à gauche. Chaque réponse est validée par l'appui sur "CR" qui ramène le curseur en début de ligne suivante. Sur la dernière ligne, la frappe de "CR" a pour effet de sortir du cadre et entraîne l'apparition d'un message 'Modifications (O ou N) ?'. L'appui sur 'O' a pour effet d'effacer ce message et de ramener le curseur à l'origine. L'opérateur peut alors effectuer les corrections nécessaires à l'aide des touches d'éditations. L'appui sur 'N' provoque l'affichage de la page 2.

#### V-1-5) Présentation de la page 2

"Cette deuxième page intitulée 'Modules actifs'demande les signaux physiologiques que le personnel soignant désire traiter (figure 5-1-3)"/VASS3/. Elle se compose d'un cadre divisé en lignes, chacune d'entre elles étant référencée par un numéro caractérisant les signaux que la machine peut analyser. A l'origine, le curseur se positionne sous le numéro 1 et l'opérateur choisit s'il désire traiter ce signal. A chaque réponse, le curseur passe à la ligne suivante. Pour chaque fonction, l'inversion vidéo se porte automatiquement sur l'option choisie. Lorsque

l'opérateur a choisi son traitement, l'appareil demande comme pour la première page si une modification s'avère nécessaire. Si tel est le cas le curseur pointe sous le premier numéro et l'opérateur peut effectuer les corrections voulues. Lorsque tout est correct, la machine passe à la page 3.

1	P.A.P. ou T.A.
2	Pression, Volume, CO <sub>2</sub>
3	E.C.G.



Figure 5-1-3

#### V-1-6) Présentation de la page 3

La troisième page affiche les seuils des alarmes. Les modifications de ces seuils se font de la même manière que pour la page de configuration du système (figure 5-1-4).

En fin de page, et après apparition du message désormais traditionnel, l'appui sur 'N' valide la page 3 et met un terme à l'initialisation de l'appareil. Le système commence son traitement et se place en attente d'une commande de l'opérateur. En bas de l'écran, huit pavés indiquent les rôles dévolus à chaque touche de fonction. L'emploi des commandes ne pose pas de problème de mémorisation car elles sont assez simples: edit, menu, text...

Les réponses aux trois premières pages marquent la fin de l'initialisation de l'appareil. Celui-ci est

alors prêt pour répondre aux commandes afin de faciliter l'établissement d'un diagnostic.

Alarmes CO <sub>2</sub>	
PET max (mmHg) :	
PET min (mmHg) :	
Alarmes cardiaques	
FC max :	
FC min :	



Figure 5-1-4

#### V-1-7) Description des commandes

##### V-1-7-1) INIT

Cette commande provoque l'affichage d'un message 'Initialisation (O ou N) ?'. Une réponse positive réinitialise l'appareil c'est à dire que toutes les données inscrites dans toutes les pages ainsi que les résultats issus des traitements effectués par la machine sont perdus.

##### V-1-7-2) MENU

'Menu' permet de visualiser les pages écrites par l'opérateur sans pouvoir les modifier.

V-1-7-3) PREC et SUIV

Ces touches ne sont actives qu'avec la commande ci-dessus. Elles permettent une visualisation par rapport à la page en cours de la page précédente ou suivante.

V-1-7-4) TEXT

la commande 'TEXT' permet de rajouter des commentaires dans la partie remarque de la page 1. Chaque commentaire tapé au clavier est automatiquement précédé de l'heure de son entrée.

V-1-7-5) SIGN

Cette commande permet le démarrage ou l'arrêt du traitement d'un signal sélectionné dans la page 2.

V-1-7-6) ALAR

Lorsqu'une alarme survient, l'opérateur peut en connaître immédiatement la nature en visualisant la page 3. Il peut également rappeler cette page pour modifier les seuils d'alarme établis lors de la phase d'initialisation.

V-1-7-7) EXAM

C'est la commande principale du système: elle donne accès à la page 4 qui propose différents types d'examen.

V-1-8) Présentation de la page 4

Cette page permet de sélectionner l'examen que l'opérateur souhaite effectuer. Pour cela, quatre examens lui sont proposés. Il suffit de répondre par le numéro correspondant au choix désiré.

1	Examen simultané des situations (échelle temporelle)
2	Examen simultané des situations (échelle spatiale)
3	Bilan pour un signal isolé
4	Visualisation des signaux

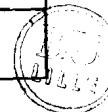


Figure 5-1-5

Examen N°1: examen simultané des situations (échelle spatiale).

Cet examen visualise en temps réel les graphes de situation regroupant les signaux d'origine cardiovasculaire et respiratoire. Les figures 5-1-6 à 5-1-8 représentent trois exemples de graphes obtenus pour les signaux d'origine cardiovasculaire.

La figure 5-1-6 est le modèle type des signaux d'origine cardiovasculaire. Chaque segment de droite reliant les branches représentatives d'un paramètre forment un triangle équilatéral. Ce triangle régulier permet d'affirmer qu'aucune anomalie n'est à signaler. La valeur de la P.A.P. est normale de même que le cycle et la fréquence cardiaque.

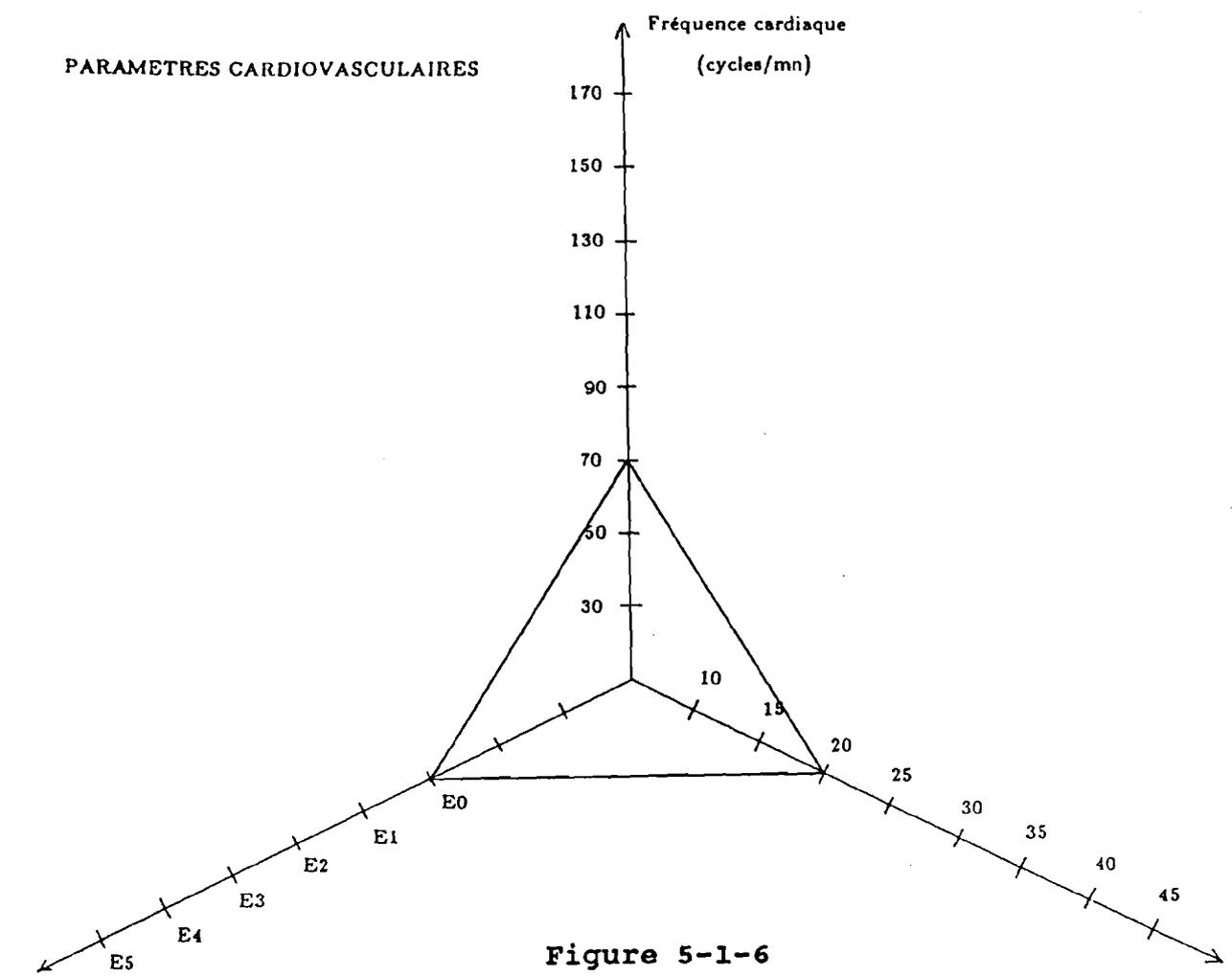


Figure 5-1-6

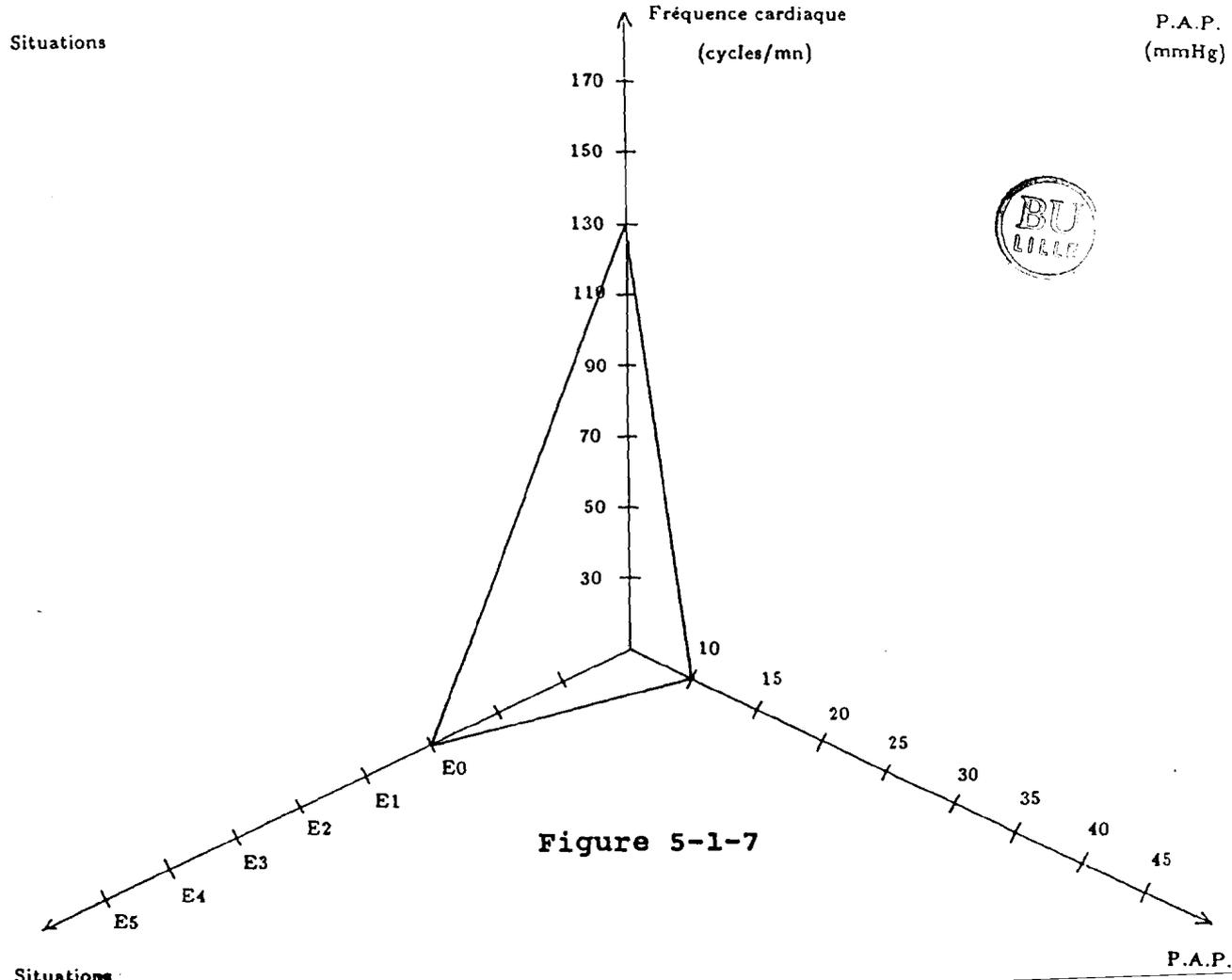


Figure 5-1-7



La figure ci-dessus est un cas typique d'hypovolémie (désadaptation contenant/contenu). Le coeur essaie de récupérer son débit en augmentant sa fréquence. La forme du triangle est révélatrice de cette pathologie.

La figure suivante est la suite logique de l'exemple précédent. Après un temps variable, le coeur peut commencer à présenter des troubles du rythme.

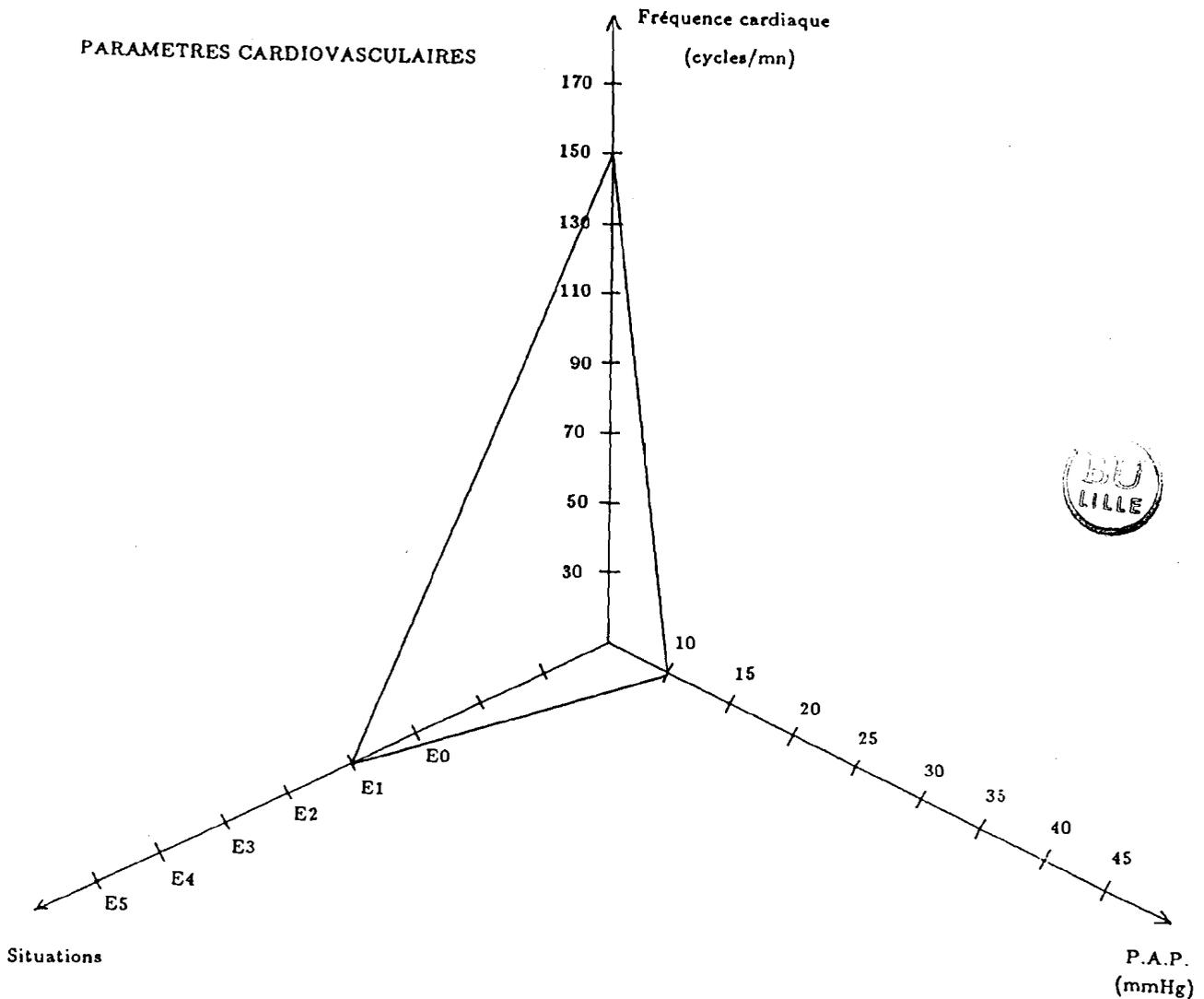


Figure 5-1-8

Le graphe des signaux respiratoires est représenté sur la figure 5-1-9. Chaque segment de droite reliant les points de chaque branche appartenant à l'échelle normale forme un pentagone régulier. - Cinq branches apportent certainement des difficultés d'interprétation d'un tel graphe, c'est pourquoi il est préférable de

représenter les signaux d'origine cardiovasculaire et respiratoire sur deux graphes distincts. Cela peut tout de même permettre d'établir des corrélations entre eux et ainsi pouvoir mettre en cause telle fonction plutôt qu'une autre en cas d'anomalie. Par exemple, si le rythme cardiaque se ralentit, cela aura ultérieurement une répercussion sur le paramètre  $VCO_2$  qui baissera, car le coeur risque de ne pas pouvoir conserver longtemps le même débit sanguin. Autre exemple: Si la fonction cardiaque est stable et que l'on constate une augmentation ou une diminution de pression du gaz carbonique expiré, il faut tenir compte de l'état d'autres paramètres avant d'émettre un diagnostic. Si  $VCO_2$  augmente et DuV également, c'est un signe d'amélioration de l'état des poumons du patient.

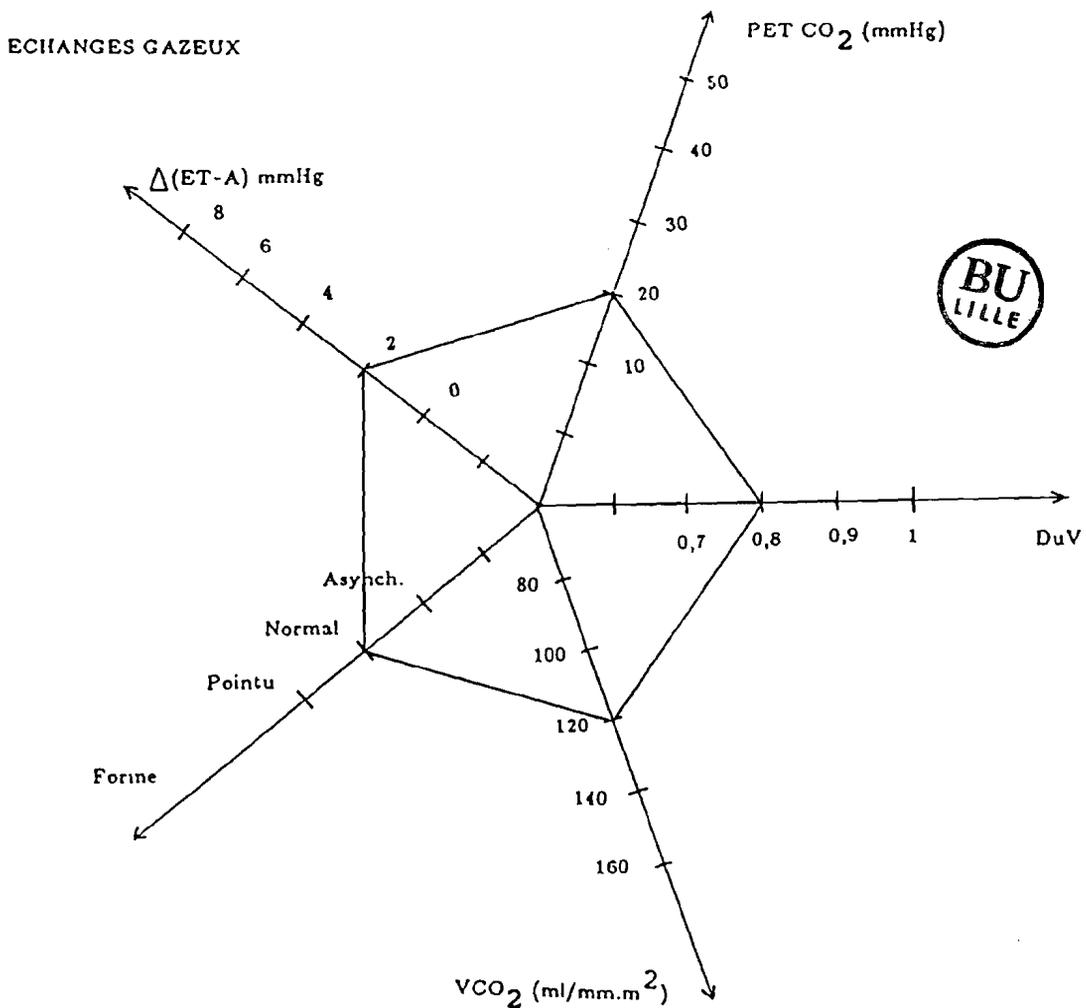


Figure 5-1-9

Examen N°2: Examen simultané des situations (échelle temporelle).

Cet examen permet d'établir d'éventuelles corrélations entre les signaux. Une fenêtre initiale s'affiche à l'écran, et l'opérateur prend la main pour modifier cette fenêtre à l'aide des six touches numériques '1' à '6'.

Les touches '1' et '6' concernent la fenêtre supérieure, celle des heures. Un examen sur un maximum de six heures peut être effectué.

La touche '1' provoque le déplacement d'une heure dans le passé (maximum 24h) et la touche '6' d'une heure dans le présent (minimum 0h).

Les touches '2', '3', '4' et '5' concernent la fenêtre inférieure, celle des quarts d'heure.

Les touches '2' et '5' agrandissent la fenêtre d'observation d'un quart d'heure: touche '2' dans le passé, touche '5' dans le présent.

Les touches '3' et '4' diminuent la fenêtre d'observation d'un quart d'heure: touche '3' dans le présent, touche '4' dans le passé.

Lorsque la fenêtre a été positionnée selon les souhaits de l'opérateur, l'appui sur 'CR' provoque l'affichage d'une liste des graphes de situations qu'il souhaite visualiser. Ce choix étant fait pour un maximum de trois graphes, l'écran passe en mode graphique et visualise les courbes de situation en indiquant une échelle horaire fonction de la période sélectionnée ainsi que les intitulés des paramètres.

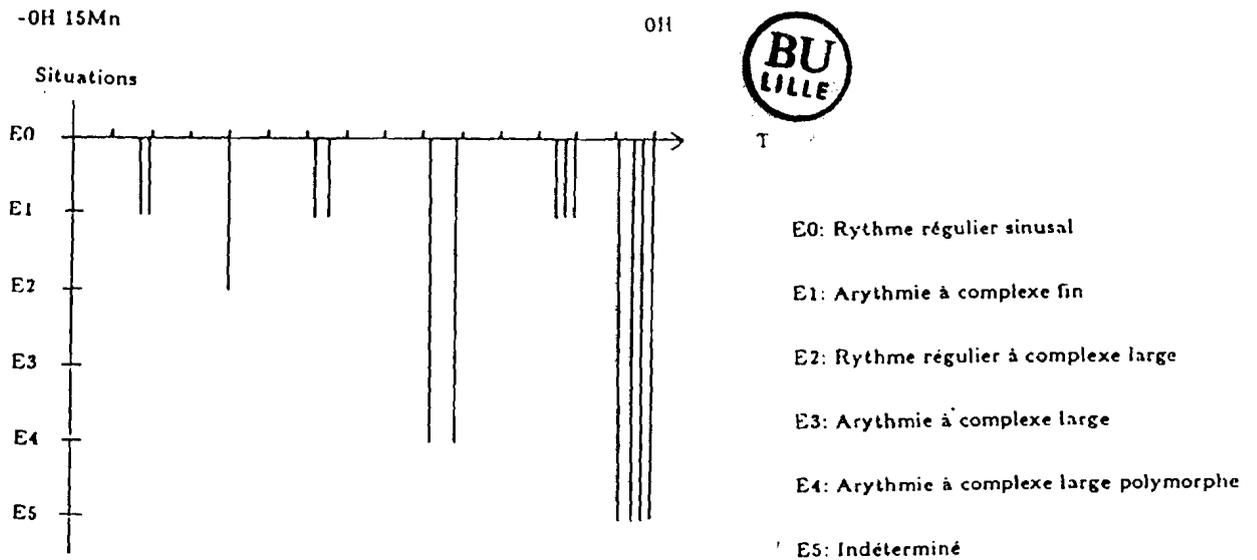


Figure 5-1-10

La figure 5-1-10 représente ce que peut être la trace d'un quart d'heure relative à l'électrocardiogramme précédent le quart d'heure en cours d'élaboration. Les troubles du rythme étant classés dans un ordre de gravité croissante, cet examen permet d'avoir une appréciation sur l'évolution du myocarde.

Examen N°3: Bilan pour un signal isolé.

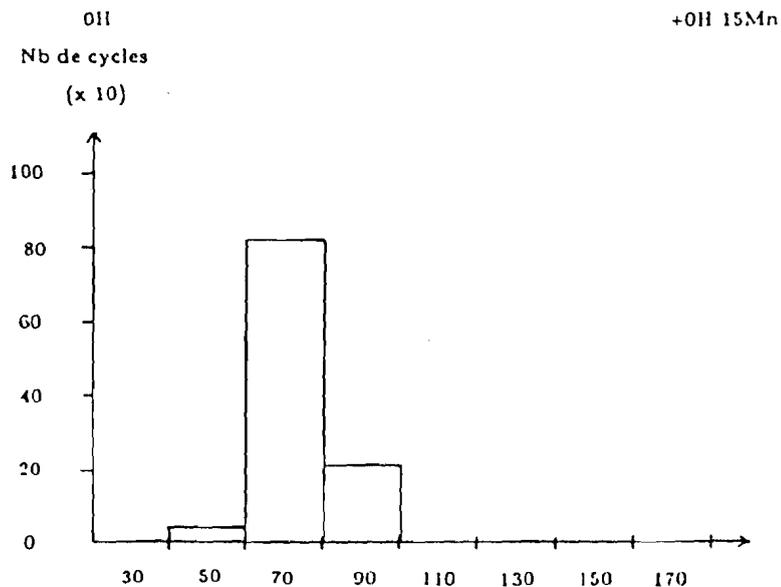


Figure 5-1-11

L'appareil visualise la fenêtre initiale décrite dans l'examen n°2. Trois histogrammes choisis dans la liste des signaux cardiovasculaires ou respiratoires peuvent être visualisés simultanément après sélection de la période horaire. La figure 5-1-11 est représentative de ce type d'examen. Elle visualise le nombre de cycles cardiaques du quart d'heure en cours de construction. On peut affirmer que le rythme cardiaque est normal car la moyenne de l'histogramme se situe autour de la fourchette de 70 battements par minute.

#### Examen N°4: Visualisation des signaux.

C'est le mode moniteur classique. Une liste de signaux à visualiser en temps réel est proposée à l'opérateur. La trace d'un maximum de deux signaux peut être suivie. Les points visualisés proviennent de l'échantillonnage du signal. Un point sur quatre est affiché ce qui peut permettre de faire un zoom du signal. On réalise le gel de la trace par appui sur la touche 'G'. On peut alors zoomer une partie du signal grâce au déplacement d'un curseur manipulé avec les touches 'l' (à gauche) et '6' (à droite). Une nouvelle frappe sur la touche 'G' permet de revenir au mode normal.

#### V-1-9) Conclusion

Ainsi s'achève la description du "mode d'emploi" et des possibilités d'aide au diagnostic que peut apporter ce système. La deuxième partie de ce chapitre est consacrée à l'étude du logiciel permettant la mise en oeuvre des différents dialogues opérateur/système qui viennent d'être décrits.

## V-2) Présentation générale

### V-2-1) Conception modulaire

Le système étant modulaire et extensible, il a fallu définir les règles permettant de respecter ces critères afin que toute adjonction ou retrait de module n'entrave pas le fonctionnement de l'appareil.

- Règles sur les programmes: lorsque plusieurs programmes indépendants sont exécutés simultanément (logiciel d'application), ils possèdent chacun leurs propres zones de données globales accessibles par le logiciel d'exploitation. Un programme d'exploitation est constitué de un ou plusieurs processus partageant la même zone de données locales.

- Règles sur les unités: les unités constituent des services offerts au logiciel d'exploitation par le biais d'une interface (mémoire globale) ne possédant que des déclarations de variables, les paramètres d'affichage.

- Règles sur les extensions: les extensions représentent un module de code adjoint aux autres modules d'application. Elles signalent leur présence, tout comme les autres modules par l'intermédiaire d'une unité d'interface.

- Règles sur les bibliothèques: les bibliothèques sont les logiciels qui permettent de faire tourner les extensions. Les liens entre le module d'extension et le système d'exploitation s'établissent de la manière suivante: le premier appel nécessitant l'établissement d'un lien est dérouté vers une routine du système qui fournit l'adresse de l'unité d'interface du module. Les appels suivants fonctionnent pratiquement comme des appels de procédure ordinaire. Les bibliothèques sont utilisées notamment pour le

calcul des coordonnées d'affichage des paramètres à l'écran.

### V-2-2) Archivage et accès aux données

Ce système devant mémoriser les informations issues du traitement des signaux sur vingt-quatre heures, c'est un module de gestion de fichiers qui en assure la lecture et l'écriture en fonction d'une demande venant de l'extérieur ou de la mémoire globale.

Ce module permet la création et la destruction de fichiers ainsi que l'accès aux pages de ceux-ci. Les fichiers sont créés lors du lancement de l'appareil (boot). En ce qui concerne les fichiers des modules d'extension, ils sont créés lors du premier appel qui lie ce module à la machine. Dans tous les cas, les noms de fichiers sont attribués, et toutes les pages à l'intérieur du fichier sont identifiées de façon unique. Ce système d'accès en mode page simplifie la gestion du système de stockage et de lecture des données mais nécessite la création d'un module de gestion globale des fichiers. Ce module contient également une fonction de recouvrement qui permet d'écraser les données datées de plus de vingt-quatre heures pour les remplacer par les nouvelles données issues de l'examen en cours.

### V-2-3) Partage du processeur

#### V-2-3-1) Exécution des programmes

Tous les programmes utilisant le terminal sont gérés par un sémaphore noté 'Géné', qui leur attribue la ressource permettant leur exécution, le microprocesseur.

Rappelons qu'un sémaphore est constitué d'un compteur à valeur entière noté C, d'une file d'attente notée F et ne peut être manipulé que par deux primitives notées WAIT et SIGNAL. Soit un processus P1 qui exécute WAIT ou SIGNAL et un processus Q1

appartenant à la file d'attente, les deux primitives s'écrivent:

```
Type Semaphore_Gene=Record
  C:Integer;
  F:File_Attente;
End.

Procedure Init_Gene (Var S: Semaphore_Gene);
Begin
  S.C := 1;
  S.F.Tete:=Nil;
  S.F.Queue:=Nil;
End.

Procedure Wait_Gene (Var S: Semaphore_Gene);
Begin
  S.C := S.C,1;
  If S.C < 0 Then
  Begin
    état(P1):= Bloqué;
    entrer(P1,S.F);
  End;
End.

Procedure Signal_Gene (Var S: Semaphore_Gene);
Begin
  S.C := S.C+1;
  If S.C =< 0 Then
  Begin
    sortir(Q1,S.F);
    état(Q1):= Actif;
  End;
End.
```

Seule l'exécution de la primitive Wait est susceptible de bloquer un processus. Il ne peut être réveillé que s'il est en tête de la file d'attente et qu'à ce moment, un autre processus exécute Signal. La file d'attente est du type premier entré, premier sorti, sans priorité. On élimine ainsi le risque de

voir un processus stagner au fond de la file sans jamais être servi.

Le module Ms étant monoprocesseur, l'exclusion mutuelle est réalisée sur ce sémaphore en inhibant les interruptions pendant l'exécution des primitives Wait et Signal.

#### V-2-3-2) Modèle de file d'attente

Lors de l'appel d'un processus, que ce soit un appel séquentiel ou aléatoire, la file d'attente ne reçoit que la valeur de l'adresse du bloc de contexte du processus.

Ceci justifie les deux règles suivantes:

- Tous les programmes gérés par un sémaphore se terminent par une instruction de saut à la primitive Signal.
- Tous les passages de paramètres s'effectuent par l'intermédiaire de la zone mémoire allouée aux blocs de contexte de processus.

La figure 5-2-1 représente un état de la file d'attente.



Figure 5-2-1

L'appel d'un nouveau processus commence par l'exécution de la primitive Wait et donc l'écriture de

l'adresse de son bloc de contexte dans le premier élément vide de la file d'attente.

La fin de l'exécution d'un processus fait appel à la primitive Signal, et donc récupère l'adresse du bloc de contexte du processus situé au sommet de la file d'attente et la décale d'un élément. L'élément entrant est nul. Si la pile est vide, le processeur retourne à sa tâche de fond. Son réveil est effectué sur l'évènement d'interruption.

Ce sémaphore constitue le noyau principal du module Ms. Les paragraphes suivants s'attachent à montrer le fonctionnement des mécanismes de communications des échanges entre processus et, entre processus et ressource commune, la console.

#### V-2-3-3) Synchronisation temporelle

La synchronisation temporelle repose sur l'utilisation d'un compteur interne cadencé par un oscillateur à quartz. Ce compteur délivre une impulsion tous les dixièmes de seconde permettant de déclencher une interruption. Le programme d'interruption permet d'activer les processus suivants:

- P1: lecture des situations
- P2: construction du graphe des situations
- P3: détection des alarmes
- P4: incrémentation de l'horloge
- P5: Gestion du processus de transfert des résultats
- P6: Gestion du sémaphore 'Géné'

Tous les processus sont exécutés séquentiellement avec une particularité pour le cinquième qui ne doit être actif que tous les quarts d'heure. Un registre compteur est utilisé en décrémentation pour autoriser l'exécution du processus P5. Sa valeur initiale est un quart d'heure. Quand ce registre a atteint la valeur nulle, P5 est exécuté.

Un échancier prend en charge la synchronisation temporelle. Il est constitué par une file d'attente contenant le descripteur de processus. Ce descripteur se compose de l'identité du processus et de son heure d'exécution.

```
Type Dec_Proc: Struc
    Nom: P5;
    Heure: Integer;
End;
```

Dans le cas de figure de cet échancier, il n'y a qu'un processus en attente. L'interruption Horloge active le processus suivant.

```
Procedure Int;
    Var ProceD: Dec_proc;
        Time: Integer;
Begin
    P1;
    P2;
    P3;
    P4;
    Time:= ProceD.Heure;
    Time:= Time-1;
    If Time:=0 Then
    Begin
        P5;
        Time:= 900;
    End;
    ProceD.Heure:= Time;
    P6;
End.
```

Cette procédure est relativement simple. Il n'en serait pas de même si plus de deux processus nécessitaient d'être activés sur des intervalles de temps différents.

Le processus P4 gère l'heure.

```
Program P4;  
Type  
  Heure_Debut:= 00 H 00 MN 00 S;  
  Heure_Reelle:= 00 H 00 MN 00 S;  
Begin  
  Heure_Debut:= Heure_Debut+ 1;  
  Heure_Reelle:= Heure_Reelle+ 1;  
  Mise_Heure;  
End.
```

Toutes les secondes, ce processus incrémente les variables `Heure_Reelle` et `Heure_Debut`. L'affichage a lieu toutes les secondes.

Le fonctionnement des autres processus s'intégrant dans la communication entre le module Ms et les modules Mi est repris au chapitre V-2-4-4.

#### V-2-4) Description des échanges

##### V-2-4-1) Communication Ms -> Console

Le terminal est relié au module Ms par une liaison série. Concrètement, sur la carte Ms, deux circuits intégrés: compteur et série se chargent de la gestion de cette ligne. Le circuit série travaille en interruption avec le microprocesseur. La ligne série elle-même est du type cinq fils: émission, réception, masse et les deux lignes de handshaking. Cette liaison est chargée de véhiculer tous les codes de contrôle constitués de plusieurs caractères, ainsi que les chaînes de caractères propres à l'affichage. Les codes de contrôle ou 'séquences escape' d'une longueur minimum de quatre caractères déterminent la position du curseur, la couleur de l'affichage, le passage du mode texte en mode graphique...

Il est aisément concevable qu'il ne faut en aucun cas interrompre l'envoi d'une série de caractères prédéfinis et reconnaissables par le terminal pour exécuter une autre action par d'autres caractères. Il en résulte un affichage ambigu dû à une mauvaise

interprétation des codes de contrôle non conforme aux souhaits du programme.

Cette situation peut se produire lors de l'affichage de l'heure à la console. Le processus d'affichage est activé toutes les secondes par interruption et doit envoyer une chaîne de caractères de contrôle contenant la position du curseur, la couleur et les caractères propres à l'affichage. Le déclenchement de ce processus peut survenir pendant l'envoi d'une chaîne de caractères vers la console par un processus quelconque et perturber l'affichage. Il est donc nécessaire de gérer l'affichage de l'heure par le sémaphore 'Géné'.

Ce processus envoie vers la console une chaîne de caractères au format suivant (figure 5-2-2):

```
ESC,'7',ESC,'31m00H 00M 00S',ESC,'8',0
```

Figure 5-2-2

Lorsque le processus P4 est exécuté dans le programme d'interruption, il parcourt la séquence suivante:

```
Procédure Mise_Heure;  
1)   If Semaphore_gene.C >= 0 Then  
      Begin  
2)   Ecriture_heure;  
      P2 := Affichage_heure;  
3)   Wait_gene(Var S:Semaphore_gene);  
      End.
```

1) Teste si l'exécution du processus d'affichage a eu lieu.

2) Programme d'écriture des caractères dans la file d'attente réservée à cet effet.

3) Appelle à la procédure Wait qui met le programme d'affichage en file d'attente.

La libération de la console est effectuée par le processus courant en retournant sur la procédure Signal à la fin de son travail.

V-2-4-2) Communication Console -> Ms

Les commandes de l'opérateur sont transmises au module Ms via une liaison série gérée par interruption. Les caractères issus du clavier sont stockés dans un buffer au format suivant (figure 5-2-3):

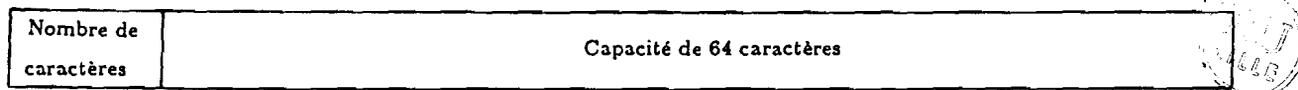


Figure 5-2-3

Le premier octet indique le nombre de caractères présents dans le buffer. Une procédure d'analyse de commande se charge de leur exécution et de la gestion du buffer. Les commandes proviennent des touches de fonctions programmées pour envoyer deux caractères non accessibles à partir du clavier standard afin d'éviter toute confusion dans l'utilisation du système. Concernant l'entrée de paramètres par l'opérateur, la procédure d'analyse de commande est dessaisie de ces fonctions au profit du programme qui gère les pages nécessitant ces entrées.

Il existe neuf types de commandes accessibles par les touches de fonctions. Elles permettent l'accès aux différentes pages de la console et donc aux fonctions offertes à l'utilisateur. Ces fonctions sont numérotées CP1 à CP9. Après reconnaissance d'une commande, la fonction correspondante est mise en file d'attente pour être activée.

V-2-4-3) Communication C.A.D. -> Mi

L'acquisition des échantillons est réalisée par les interfaces parallèles des modules Mi qui commandent l'échantillonnage de la carte C.A.D. selon leur fréquence propre, et la réception des échantillons. Un timer délivre une impulsion de période  $T_e$  avertissant le convertisseur de la carte C.A.D. qu'il peut procéder à l'échantillonnage par la ligne "Start". Ce travail étant fait, le convertisseur avertit le module Mi sur sa ligne "Data Ready" qu'il peut recevoir l'échantillon.

Le matériel de communication est le même pour les modules traitant de la pression artérielle et des signaux respiratoires: un seul convertisseur doit réaliser l'échantillonnage de trois signaux pour le module P.V.C. La figure 5-2-4 représente l'acquisition des signaux pour ce dernier. Quant aux modules traitant de l'E.C.G. et de la pression artérielle, un convertisseur A/D se charge de leur acquisition.

V-2-4-4) Communication Mi -> Ms

La communication entre les modules Mi et le module de synthèse s'effectue à travers la mémoire tampon B.A.L. Neuf types de dialogues, classés en trois catégories, peuvent s'établir entre les modules.

a) La première est relative au démarrage des modules Mi.

b) La seconde regroupe les fonctions séquentielles du système d'exploitation:

- Transfert des informations tous les quarts d'heure.
- Lecture des situations et détection des alarmes.

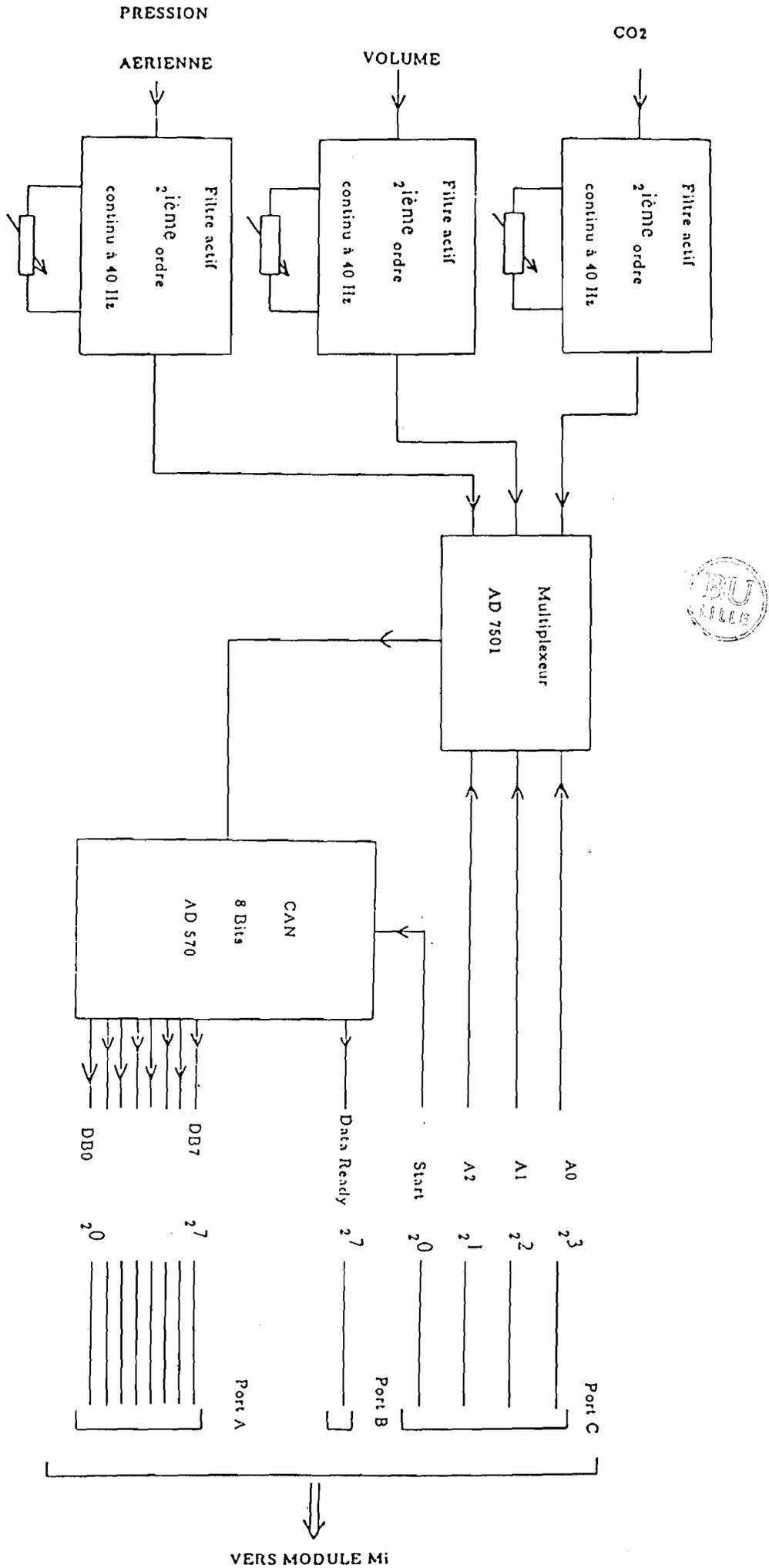


Figure 5-2-4

VERS MODULE Mi

b) La troisième regroupe toutes les fonctions aléatoires déclenchées sur commande de l'opérateur:

- Changement de paramètres (CP1).
- Bilan pour un signal isolé (CP2).
- Examen simultané des situations en échelle temporelle (CP3).
- Visualisation des signaux (CP4).
- Examen simultané des situations en échelle spatiale (CP5).
- Visualisation des histogrammes en cours pour un signal (CP6).
- Visualisation page précédente (CP7).
- Visualisation page suivante (CP8).
- Fichier du patient (CP9).

Toutes les communications utilisent le système du producteur au consommateur. Les tâches vont communiquer entre elles au moyen de messages. Les demandes de requêtes proviennent principalement de l'échéancier: toutes les secondes pour les situations, tous les quarts d'heure pour les histogrammes. Un message est transmis afin de demander un service au producteur, lequel renvoie un message pour indiquer que l'opération a eu lieu. Cet échange se déroule en quatre étapes: une étape de demande d'information, une seconde d'envoi de message d'une tâche à une autre, une troisième de fin d'émission, et une dernière d'accusé de réception.

#### Démarrage des modules Mi

A l'allumage du système, la première tâche effectuée par les modules Mi consiste à signaler au module Ms leur présence et leur signature par positionnement de l'indicateur DEMMI (B.A.L.). Après lecture par Ms des indicateurs, l'opérateur est alors averti par la console, du nombre et du type de signaux physiologiques qui peuvent être traités par le système. La signature sert à connaître le type de traitement et le nombre de signaux que la carte peut traiter.

En prévision d'une possible extension du système, le module Ms va tester jusqu'à huit zones prédéfinies dans la mémoire B.A.L. C'est le seul test indispensable pour le système afin que le module Ms puisse par la suite lier les bibliothèques de programme destinées à la gestion des modules Mi. Ce test ne peut renvoyer que la valeur nulle si un module Mi n'est pas connecté au système car toute la mémoire tampon est remise à zéro au lancement du système.

Le critère de modularité impose donc la création de huit zones identiques dans le module B.A.L. Ainsi, tous les paramètres sont situés à une adresse relative par rapport au début de chaque page. La carte mémoire d'une page est définie figure 5-2-5.

Les zones de transfert et de paramètres sont issues des modules Mi.

- La première zone occupe deux octets mémoire. Le premier indique la valeur de l'octet de démarrage de la carte. Il est positionné à 1 par le module Mi. Le module Ms le remet à zéro si l'opérateur souhaite exploiter cette carte. Le deuxième octet regroupe toutes les demandes d'examen venant de Ms et nécessitant un service de Mi (figure 5-2-6). Le module Mi vérifie ainsi facilement si Ms réclame d'être servi.

- Les huit derniers mots de la zone deux reçoivent les paramètres de fonctionnement du module Mi. Le premier octet est utilisé en acquittement par Mi en fonction d'une demande de transfert venant de Ms. De même pour le premier octet des zones 3, 4 et 5.

- La zone 3 reçoit les histogramme du module Mi. Un histogramme occupe 64 octets mémoire. Cette zone peut recevoir au maximum huit histogrammes.

- Les graphes de situations sont transmis par l'intermédiaire de la zone 4.

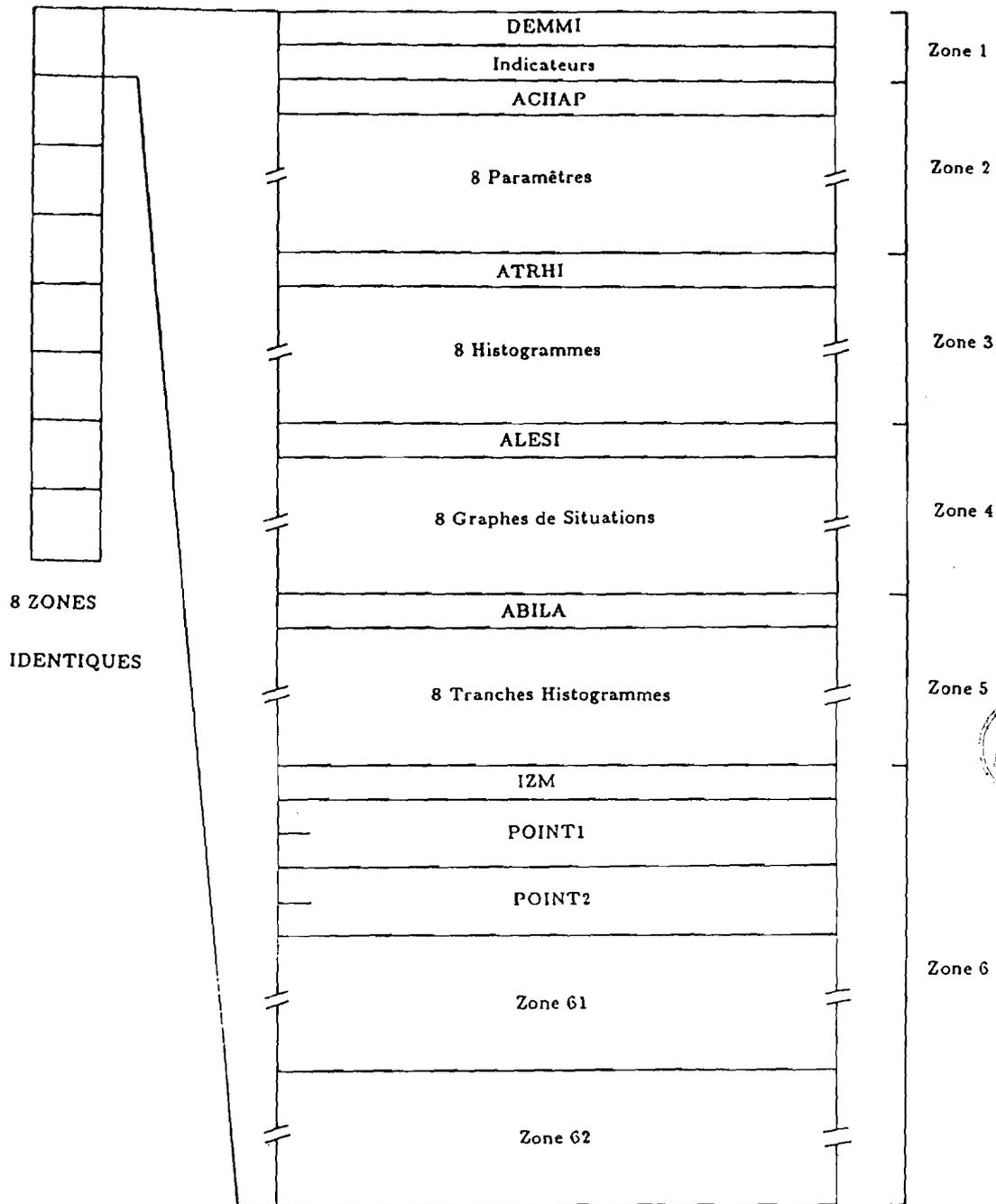


Figure 5-2-5

			DEM	LESI	TRHI	CHAP	BILA
--	--	--	-----	------	------	------	------

Figure 5-2-6

- La zone 5 est utilisée pour l'envoi des histogrammes en cours de construction.

- La zone 6 stocke les échantillons lorsque l'opérateur demande une visualisation des signaux.

De son côté, le module Ms utilise les modules de code pour remplir ces tables internes (Figure 5-2-7).

Les extensions nécessaires au module Ms pour prendre en charge les résultats des modules Mi sont situées dans la table TAB\_ADR. Chaque élément est un index pointant sur les paramètres de gestion de chaque signal. Chaque unité contient les éléments de la figure 5-2-8.

TAB\_DEM  
Demande de traitement

Module1
Module8

Traitement de 8 signaux



Figure 5-2-7

Chaque liste est constituée par les différents axes suivi de leur coordonnées d'implantation à l'écran. Le cas de la représentation spatiale est particulier car les signaux d'origine cardiovasculaires sont regroupés sur un graphe commun. Il en va de même pour les signaux d'origine respiratoire.

TAB\_ADR

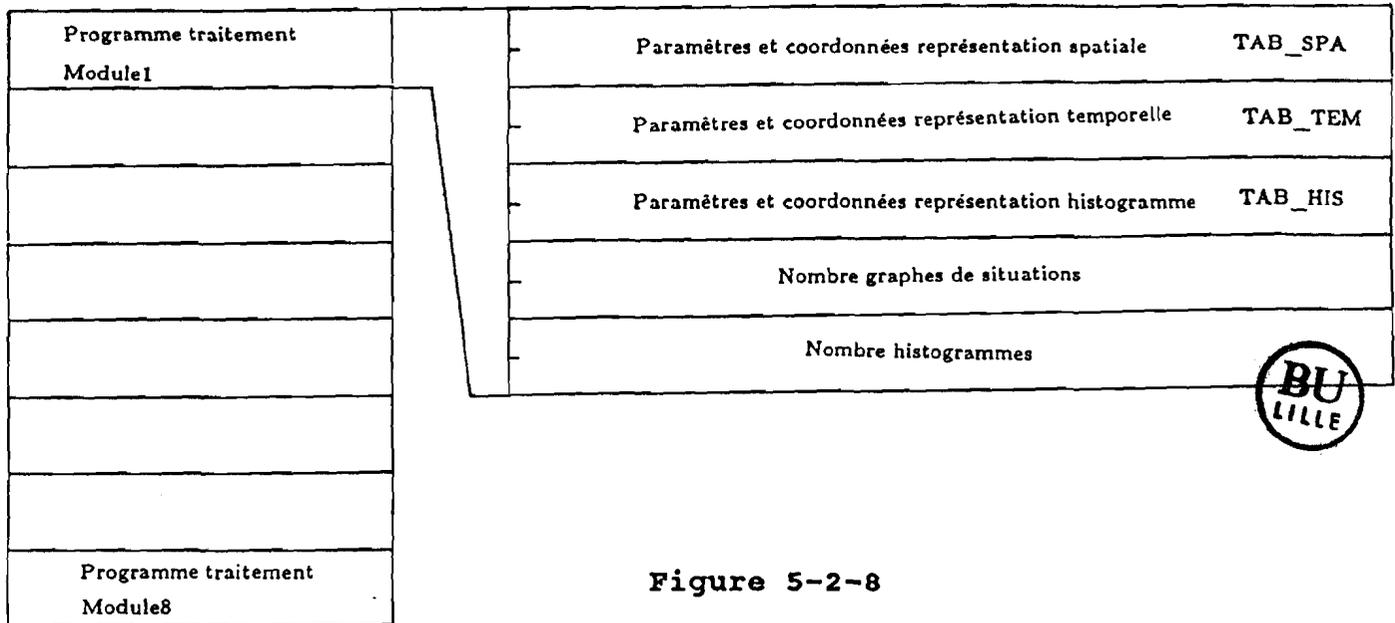


Figure 5-2-8

Transfert des résultats

Tous les quarts d'heure, le module Ms informe les modules Mi qu'ils doivent transférer le résultat de leurs traitements dans la mémoire B.A.L. afin qu'ils puissent être stockés dans la mémoire de masse. La procédure employée met en oeuvre le principe du producteur (Mi) au consommateur (Ms). Le transfert suivant est décrit pour un module Mi et est applicable pour tous les autres.

```

Program Trans_Module_Ms;
Var TRHI, ATRHI: Boolean;
Begin
1)   TRHI:=True;
2)   Repeat Until TRHI:=False;
3)   Timer XX;
4)   Repeat Until ATRHI:=True;
5)   ATRHI:=False;
End.

```

```

Program Trans_Module_Mi;
Var TRHI, ATRHI: Boolean;
Begin
6)   TRHI:=False;
7)   Trans_Resultats;
8)   Raz_Zone_Resultats;

```

9)           ATRHI:=True;  
          End.

Ces échanges fonctionnent avec la zone 3 accordée dans la mémoire B.A.L. pour chaque module Mi. La séquence d'exécution est la suivante:

- 1) Autorise le début des échanges.
- 2) Ms attend que Mi ait pris en compte la commande.
- 6) Le module Mi acquitte la commande.
- 3) Le module Ms va lire dans la table des timers le temps pendant lequel il doit rester en attente pour que le module Mi écrive les histogrammes dans la mémoire tampon: ceci afin d'éviter un test prolongé de l'étape 3 qui pénaliserait l'activité du multibus.
- 7) Le module Mi écrit les histogrammes dans la B.A.L.
- 4) Ms s'assure que Mi a terminé son travail. Il teste Atrhi jusqu'à ce que Mi l'ait positionné. La boucle Repeat Until peut être jugée inutile: c'est une sécurité au cas où d'autres modules Mi auraient eu besoin du multibus pour procéder à un changement de situation ce qui aurait pour effet de retarder le transfert en cours.
- 8) Le module Mi réinitialise sa zone de travail...
- 9) et signale que son transfert est terminé.
- 5) Le module Ms accuse réception de la part du module Mi.

Cette procédure est répétée pour chacun des modules Mi trouvé dans la table de démarrage TAB\_DEM.

Remarque: les procédures standards des systèmes utilisent une variable entière qui compte le nombre de messages présents dans la zone d'échanges. Dans le cas des transferts entre Mi et Ms, les histogrammes constituent les messages. Les modules Mi se chargent de les envoyer dans la B.A.L. en une seule fois, ce qui permet de minimiser l'occupation du multibus. D'autre

part, Ms n'est intéressé que par la fin de l'émission de Mi, sans se préoccuper de connaître la progression de son transfert. C'est pourquoi, un seul indicateur de fin de transfert est utilisé.

### Lecture des situations

Les situations des modules Mi sont lues toutes les secondes afin de détecter une défaillance d'un signal. Néanmoins, on suppose qu'un signal ne peut passer d'une situation à une autre que sur une période de temps de quinze secondes. C'est pourquoi, la construction du quart d'heure des situations s'effectue uniquement lorsque ce laps de temps est écoulé.

```
Program Sit_Module_Ms;  
Var LESI, ALESI: Boolean;  
Begin  
  LESI:=True;  
  Repeat Until LESI:= True;  
  Timer XX;  
  Repeat Until ALESI:=True;  
  ALESI:=False;  
End.
```

```
Program Sit_Module_Mi;  
Var LESI, ALESI: Boolean;  
Begin  
  LESI:=False;  
  Lecture_Situation;  
  ALESI:=True;  
End.
```

Ce programme est pratiquement identique à celui du transfert des résultats. Le nombre des situations variant d'un module à l'autre, seules diffèrent la valeur de la procédure Timer et la zone de transfert dans la mémoire B.A.L. La zone 4 du module B.A.L. est utilisée. Cette zone permet de mémoriser huit graphes de situations et huit situations par graphe.

Cette procédure est répétée pour tous les modules Mi.

La procédure suivante a pour rôle de lire les situations dans la carte B.A.L., de les stocker dans un buffer pour être analysées par le programme de gestion des alarmes et enfin de les recopier dans la zone de construction des situations.

```
Program Lectsit_Module_Ms;  
Begin  
1)   For I:= 1 To Nombre_Module_Mi  
      Begin  
2)     Lecture_Situation_Module_Mi;  
3)     Ajout_Graphe_Situation;  
4)     Detection_Alarme;  
      End;  
End.
```

- 1) La lecture s'effectue en fonction du nombre de signaux actifs trouvés dans la table TAB\_SIG.
- 2) Les situations sont lues pour chaque signal...
- 3) et ajoutées aux graphes en cours de construction.
- 4) Ensuite la procédure de détection des alarmes est activée.

#### Changement de paramètres

A l'initialisation, les modules Mi recopient dans la table qui leur est impartie en mémoire B.A.L., les paramètres par défaut entrant en compte dans leur analyse. Si l'opérateur souhaite modifier ces paramètres, le programme suivant est activé:

```
Program Para_Module_Ms;  
Var CHAP,ACHAP: Boolean;  
Begin  
1)   Changement_Parametres;  
2)   CHAP:= True;  
3)   Repeat Until ACHAP:= True;
```

```
4)    ACHAP:= False;
5)    Signal_Gene(Var S:Semaphore_Gene);
      End.
```

```
Program Para_Module_Mi;
Var CHAP,ACHAP: Boolean;
Begin
6)    CHAP:= False;
7)    Lecture_Parametres;
8)    ACHAP:= True;
      End.
```

1) C'est une procédure permettant à l'utilisateur de dialoguer avec la console pour modifier les paramètres.

2) C'est par l'intermédiaire d'une touche de validation que l'on sort de cette procédure. Le module Mi est averti par CHAP qu'il peut prendre en compte les nouveaux paramètres.

6) Acquiescement de la commande.

7) Prise en compte des nouveaux paramètres.

8) Indicateur de fin de prise en compte.

3) Ms s'assure que Mi a terminé son travail...

4) et s'en acquitte.

5) Libération du processeur.

Ce programme utilise la zone 2 du module B.A.L. et permet de mémoriser huit paramètres par signal.

#### Bilan pour un signal isolé

C'est un programme qui met en jeu un dialogue entre le module Ms et la mémoire de masse. Après réception d'une commande de l'opérateur, le module Ms est chargé d'aller lire dans le module Mm les histogrammes d'un signal et de procéder à leur assemblage.

```
Program Bilan;
Begin
1)    Position_curseur (Pointeur_haut,Pointeur_bas);
2)    Lecture_Histo (Pointeur_haut,Pointeur_bas);
3)    Assemblage_Histo;
```

4) Affichage\_histo;

End.

1) Cette procédure récupère la valeur des pointeurs fixant dans le temps la recherche des histogrammes.

2) Le module Ms va lire dans la mémoire de masse les histogrammes d'un signal, déterminés par les pointeurs définis dans l'étape 1. Tous les quarts d'heure représentatifs d'un histogramme sont lus d'un bloc et stockés dans un buffer de 1,5 K octets, ceci afin de minimiser l'occupation du multibus. La moyenne des histogrammes s'effectue directement avec le buffer ce qui évite un nombre d'accès au multibus égal au nombre de quarts d'heure à assembler. Ce buffer ne peut contenir que 24 histogrammes. Si un nombre supérieur est demandé, plusieurs accès à la mémoire de masse seront donc nécessaires. Six Koctets sont suffisants pour mémoriser les histogrammes d'une situation. La faible capacité de mémoire vive sur le module Ms fixe la restriction à 1,5 Koctets. Cette procédure prend en compte le nombre d'histogrammes trouvés dans la table TAB\_HIS.

3) Le module Ms effectue la moyenne des histogrammes...

4) et les affiche à la console.

#### Examen simultané des situations en échelle temporelle

Le principe du fonctionnement est le même que pour l'examen précédent. Ce sont les graphes de situation qui sont moyennés et affichés en mode de représentation temporelle pour un maximum de trois graphes observés simultanément. Le même buffer est employé pour stockage.

Program Sit\_Temp;

Begin

Position curseur(Pointeur\_haut,Pointeur\_bas);

Lecture\_Sit(Pointeur\_haut,Pointeur\_bas);

```
Assemblage_Sit;  
Affichage_Sit;  
End.
```

### Visualisation des signaux

Ce programme est accessible sur demande de l'opérateur. Il consiste à afficher les échantillons issus des modules Mi pour visualiser la forme des signaux. Un maximum de deux signaux peut être visualisé simultanément. Les échantillons sont envoyés par les modules Mi dans la mémoire tampon et récupérés par Ms. Le programme met en jeu la zone 6 de la mémoire B.A.L. Cette zone est divisée en quatre:

- les pointeurs d'écriture et d'effacement sont mémorisés aux emplacements POINT1 et POINT2.
- Les zones 61 et 62 contiennent les valeurs des échantillons pour l'allumage et l'extinction d'un point à l'écran. Le pointeur d'écriture est initialisé au début de la zone 61 et le pointeur de lecture au début de la zone 62. Pendant leur travail, ces pointeurs sont toujours décalés de la valeur d'une zone. Arrivés en fin de zone 62, l'adresse de la première zone leur est attribué.

Le programme d'ensemble s'écrit:

```
Program Visu_Module_Ms;  
Var DEM, COORD_X: Integer;  
    IZM: Boolean;  
Begin  
1)   DEM:= 1;  
    COORD_X:= 0;  
2)   While DEM:= 1 And COORD_X < 255 Do  
    Begin  
3)     Repeat Until IZM:= True;  
4)     Effacement(^POINT2);  
5)     Affichage(^POINT1);  
6)     IZM:= False;  
7)     Test_Clavier (Dem);  
    End;
```

```
8)      If DEM:= 1 Then ATV:=1;
9)      DEM:= 0;
10)     Signal_Gene(Var S:Semaphore_Gene);
        End.

        Program Visu_Module_Mi;
        Begin
11)     While DEM:= 1 Do
            Begin
12)         Ecriture_Echant(^POINT1);
13)         POINT1:= POINT1+ 1;
14)         POINT2:= POINT2+ 1;
15)         If POINT1 And $FC:= 0 Then IZM:= True;
            End;
        End.
```

1) Ce programme indique tout d'abord au module Mi qu'il doit transférer ces échantillons. La coordonnée sur écran en abscisse est initialisée à zéro.

2) Côté Ms, le programme s'exécute lorsque l'opérateur le demande ou tant que l'affichage ne dépasse pas les limites de l'écran.

11) Côté Mi, il cesse également sur ordre de l'opérateur (DEM=0).

12) L'écriture des échantillons commence dans la zone 1.

13) et 14) Puis les pointeurs de lecture et d'écriture sont incrémentés d'une position.

15) Lorsque quatre échantillons ont été écrits, le module Mi en avertit le module Ms.

3) Le module Ms attend l'arrivée d'un échantillon.

4) Cette condition réalisée, il efface le point lu avec POINT2...

5) et affiche celui lu avec POINT1.

6) Puis il acquitte réception.

7) Cette procédure effectue une lecture du clavier pour interrompre le déroulement du programme. Dans ce cas, elle positionne DEM à zéro.

- 9) Il faut stopper le travail de Mi lorsque l'écran est plein.
- 8) Si la demande d'affichage est toujours présente, le processus de visualisation sera remis en file d'attente.
- 10) Le module Ms peut activer le processus suivant.

### Examen simultané des situations en échelle spatiale

Ce programme reprend les situations de tous les modules Mi stockés dans le buffer défini dans le programme Lect\_Sit\_Module\_Ms pour les afficher en mode de représentation spatiale en temps réel. Il permet de visualiser les situations traversées par les signaux d'origine soit cardiovasculaire soit respiratoire. La recherche des paramètres d'affichage s'effectue par consultation de la table d'adresses des bibliothèques pour chaque signal. Chaque bibliothèque contient une table d'adresses au format de la figure 5-2-9.

Après lecture du nombre des signaux à représenter, un calcul des coordonnées d'affichage est effectué sur toutes les coordonnées nécessaires à la représentation. Ensuite, les paramètres sont lus et affichés.

Ce programme fonctionne jusqu'à interruption par l'utilisateur.

```
Program Sit_Spa;  
Var SITA: Boolean;  
Begin  
1)   SITA:= True;  
2)   Lect_Sit;  
3)   Aff_Sit;  
4)   Test_Clavier (SITA);  
5)   If SITA Then ATS:=1;  
6)   SIGNAL(Var S:Semaphore_Gene);  
End.
```



Visualisation des histogrammes en cours

On retrouve dans ce programme de grandes similitudes avec les deux programmes précédents. L'exécution s'effectue en temps réel. Elle ne peut être ralentie que provisoirement tous les quarts d'heure afin de permettre le transfert des résultats. La zone 5 de la mémoire tampon est utilisée.

```
Program Histo_En_Cours_Module_Ms;
Var BILA, BILAP: Boolean;
Begin
1)   BILA:= True;
2)   Repeat Until BILAP:= True;
      BILAP:= False;
3)   Lecture_tranche_Histo;
4)   Affichage_Tranche;
      Test_Clavier(BILA);
      If BILA Then ATH:=1;
      Signal_Gene(Var S:Semaphore_Gene);
End.
```

```
Program Histo_En_Cours_Module_Mi;
Var BILA, BILAP: Boolean;
Begin
      While BILA Do
      Begin
5)     Ecriture_Tranche;
6)     BILAP:= True;
      End;
End.
```

1) Le module Ms attend que le module Mi ait écrit la nouvelle valeur issue de son traitement en cours.

5) Le module Mi écrit son nouveau résultat...

6) Et prévient le module Ms qu'il peut en disposer.

2) Le module Ms en effectue la lecture...

3) Détecte la tranche de l'histogramme concerné, l'augmente imperceptiblement sur l'écran et

emploie l'attribut clignotant afin que l'opérateur puisse l'apercevoir.

La procédure Lecture\_Tranche\_Histo fonctionne avec la zone 5 du module B.A.L. Elle est structurée de la façon suivante:

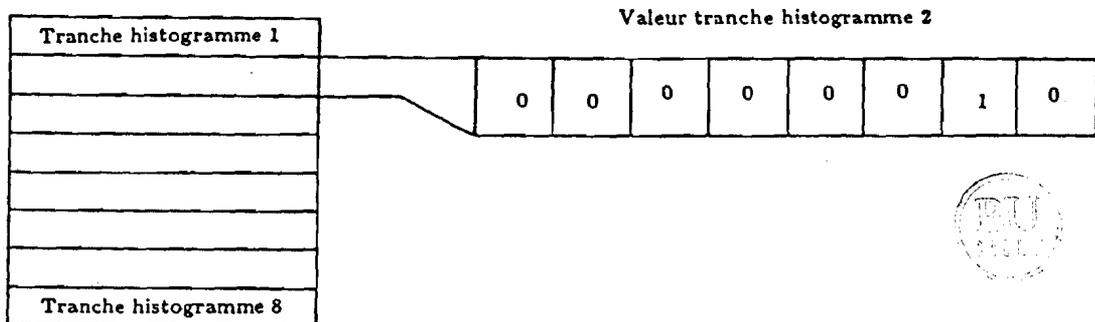


Figure 5-2-10

Chaque partie de cette zone peut mémoriser une fourchette d'histogramme. Par balayage, le module Ms peut connaître facilement quelle tranche d'histogramme doit être incrémentée. Sur la figure 5-2-10, c'est la deuxième tranche du deuxième histogramme.

Visualisation page précédente et suivante

Ces deux programmes activent l'envoi d'une chaîne de caractères à la console permettant d'accéder à la visualisation des pages précédentes ou suivantes.

V-3) Conclusion

A l'issu de ce chapitre, une question se pose: le matériel présente-t-il des faiblesses? Si c'est le cas, comment est-il possible de les éviter, et quelles sont les possibilités d'amélioration susceptibles d'être envisagées? De plus, quelles possibilités d'évolution existent pour ce type de matériel? C'est le thème du dernier chapitre.

VI) PERSPECTIVESVI-1) ModularitéVI-1-1) Interconnection de nouvelles cartes

La simulation du système est entièrement écrite en assembleur, du noyau du système d'exploitation à la présentation des résultats. Il est donc relativement facile de l'adapter et de faire des extensions. Ce langage a permis de réaliser un noyau de synchronisation de type temps réel pour échanger des événements (caractères ou messages) par l'intermédiaire d'une ressource commune.

Ce système est composé de modules: il est donc extensible. Chaque module est composé d'une partie interface et d'une partie implémentation. Ainsi, lors de l'extension du système, le code externe sera lié dynamiquement avec un autre module de référence.

VI-1-2) Problèmes logiciels

Le système constitue une station de travail évoluée et permet à un utilisateur de gérer simultanément les situations représentées par plusieurs signaux sur un même écran. C'est pourquoi l'interface homme/machine est gérée à part de façon à éliminer les effets redondants de programmation. L'implantation de nouvelles PROM doit impérativement être conforme aux tables décrites aux chapitres V-2-4.

VI-1-3) Problèmes matériels

Le multibus version de base peut recevoir un maximum de huit cartes. Tel que le système est réalisé, sept emplacements sont déjà occupés: quatre pour les modules maîtres, deux pour la mémoire de masse et la mémoire boîte aux lettres et un pour les convertisseurs A/N. Si l'on souhaite traiter d'autres signaux, il faut étendre le multibus, et donc reconfigurer le mode de gestion

parallèle. Toutefois, l'extension à un nombre élevé de modules Mi risque de saturer le multibus.

## VI-2) Evolution future

### VI-2-1) Améliorations matérielles

Actuellement, une simulation du système fonctionne. Toutes les fonctions décrites sont opérationnelles à l'exception de la page "alarmes" et des échanges pour la représentation des graphes de situations. Néanmoins, ceux-ci fonctionnent sur le même principe que la fonction "bilan pour un signal isolé". Ils sont donc implantables.

Le choix des matériels a été judicieux, à l'exception de la console LYME 6000. En effet, ce terminal est relié au module Ms par une liaison série fonctionnant à 9600 Bauds. Cette vitesse de transfert des informations est suffisante pour tous les types d'examen, sauf pour la visualisation des signaux. C'est pourquoi le module Ms ne prend qu'un point sur quatre pour des impératifs de vitesse de transmission sur la ligne série. Pour une fréquence d'échantillonnage de 50 Hz, l'affichage et l'extinction d'un point nécessite l'envoi de 30 caractères toutes les 80 mS soit 360 caractères en 960 mS. On constate donc qu'on ne peut afficher au maximum que deux signaux en ne prenant qu'un échantillon sur quatre. Le signal de l'E.C.G. étant échantillonné à 200 Hz, sa visualisation dans ces conditions n'est pas représentative de son évolution.

L'idéal serait l'utilisation d'une carte graphique au format multibus d'une résolution sensiblement égale à celle de la console LYME 6000, à laquelle on puisse connecter un moniteur vidéo couleur.

Dans ce cas, toute la gestion vidéo serait à repenser, notamment en ce qui concerne les représentations graphiques avec ré-écriture de routines de tracés de segments et de remplissage /MERI, PELE/.

VI-2-2) Conclusion

Les problèmes matériels et logiciels ayant été exposés, quelles perspectives d'avenir peuvent être envisagées pour ce système? Tout naturellement, on pense à son implantation dans un service autre que la réanimation.

VI-3) Proposition pour l'implantation dans un service de traumatologie d'urgence

La première préoccupation d'un tel service est de connaître l'existence d'un choc hémorragique sur un malade. Il est donc intéressant de pouvoir adjoindre au système une carte traitant le signal de tension artérielle. Rappelons que le système de base ne peut traiter que l'un des deux signaux de tension artérielle ou de pression artérielle pulmonaire. Pour un service de réanimation, la connaissance de ce dernier signal est également intéressante. Une carte d'extension se limitera donc au traitement de la T.A. L'intérêt principal de l'adjonction d'une telle carte consiste à pouvoir établir rapidement le diagnostic de choc hémorragique. Des exemples de représentation spatiale pour les signaux d'origine cardiovasculaire pourraient être les suivants:

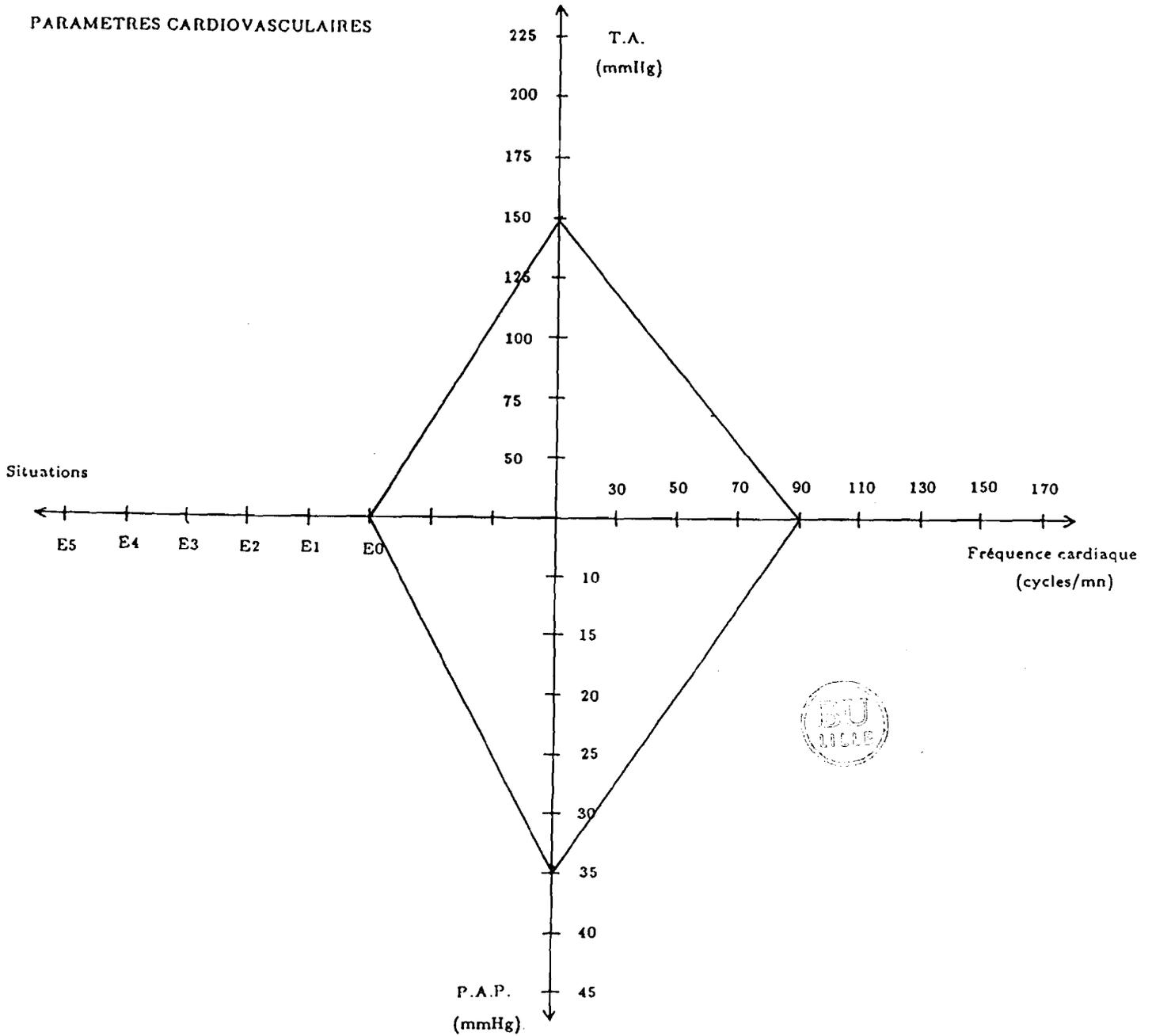


Figure 6-1

On peut interpréter cette figure de la façon suivante: le malade présente une hypertension artérielle systémique et une hypertension pulmonaire bien tolérées.

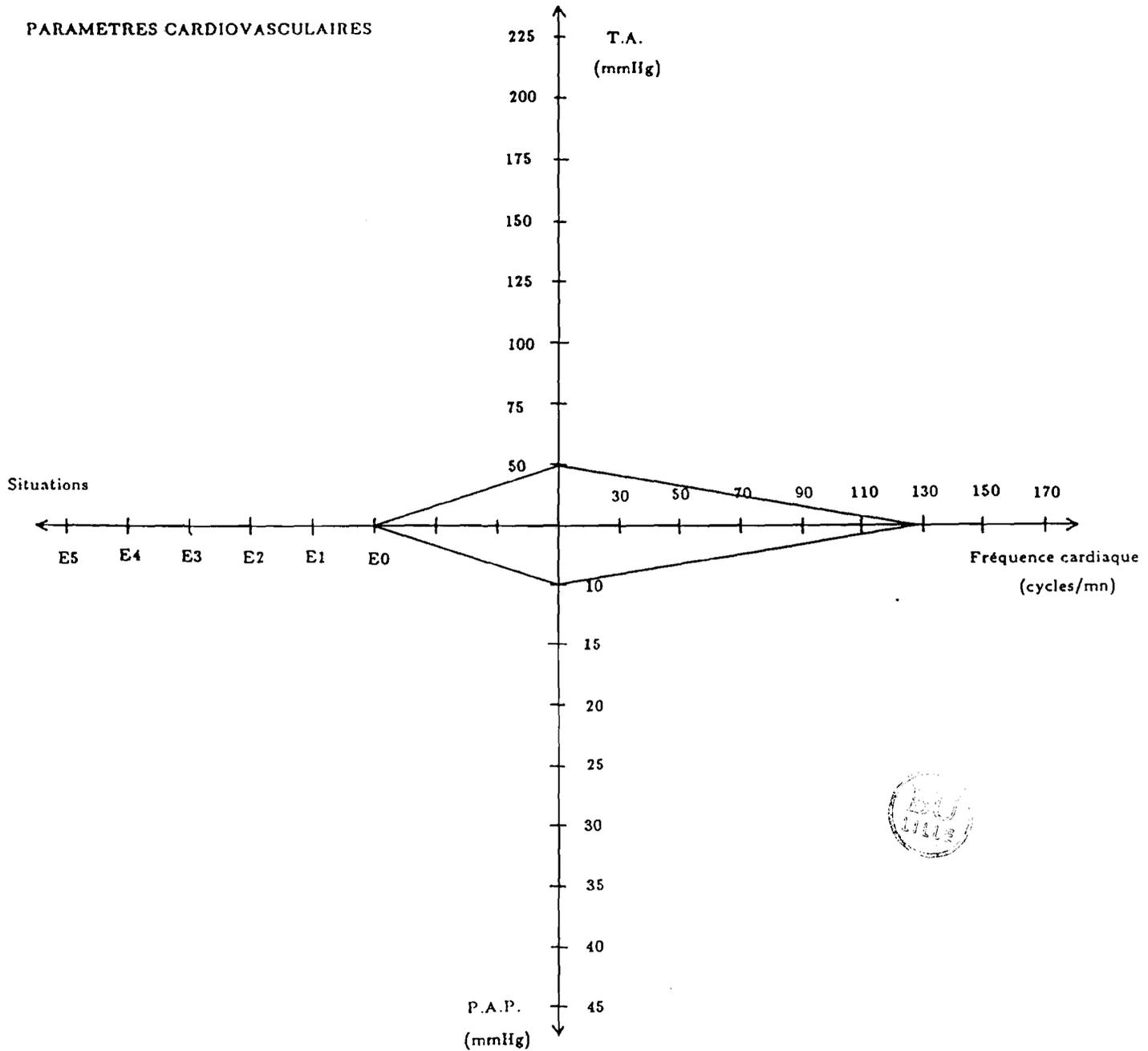


Figure 6-2

Sur cette figure, la P.A.P. et la T.A. sont très basse: la situation de ce patient est compatible avec une hypovolémie ou un choc hémorragique.

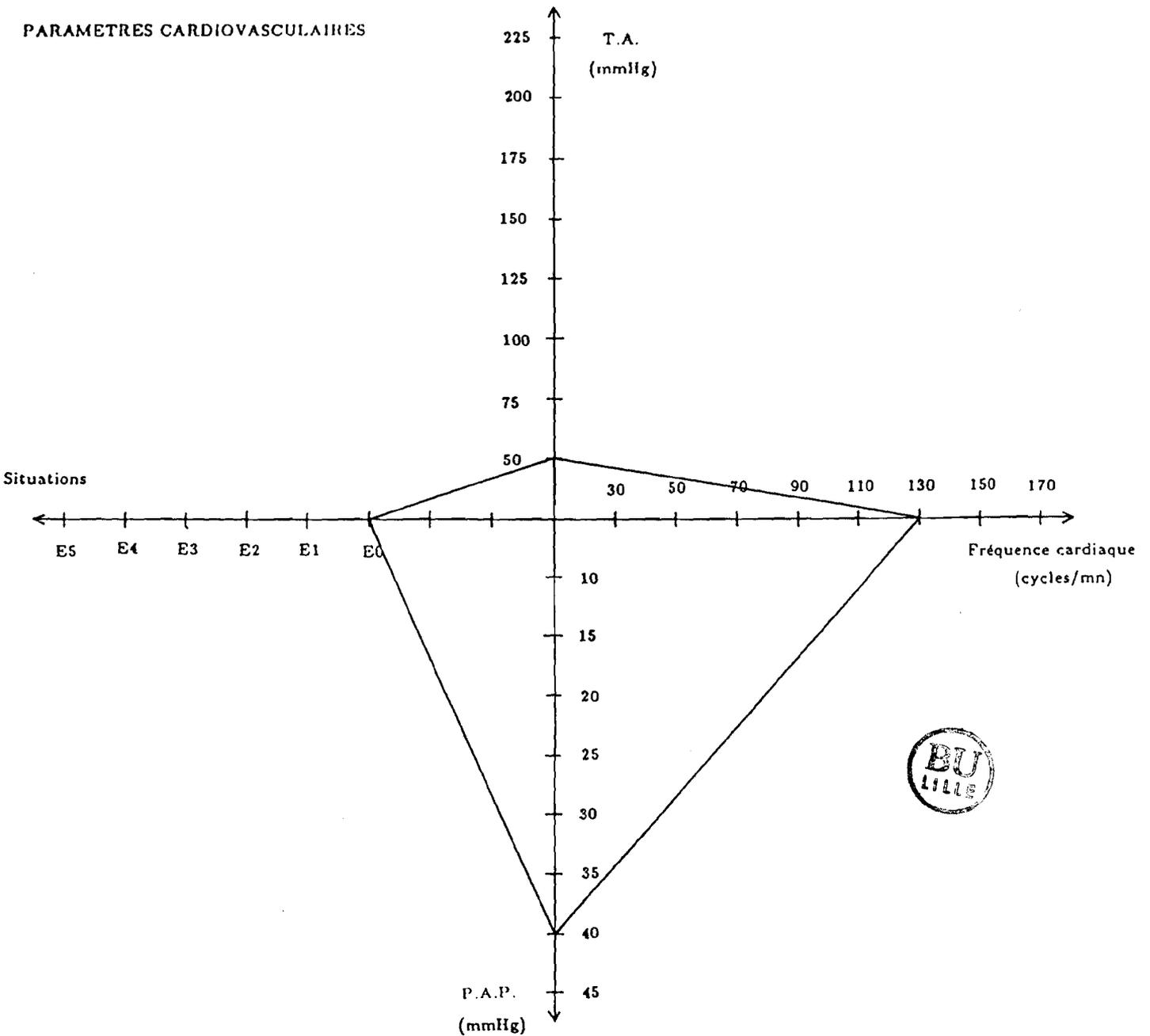


Figure 6-3

Tension artérielle basse et pression artérielle pulmonaire haute: cette situation est compatible avec une embolie pulmonaire

VI-4) Conclusion

D'autres services seraient certainement intéressés par un tel système. Avec un cahier des charges propres a chacun, la modification de système est possible si l'on respecte toutefois les règles énoncées au chapitre V.



## CONCLUSION GENERALE

L'appareil qui vient d'être décrit pourrait donner l'impression de ne pas être autre chose que la synthèse de systèmes déjà existant (Monitec, H.P., Advantage, Neuroscience, Datex Instrumentation...). Or, ceux-ci ne permettent pas au personnel soignant de pouvoir établir un diagnostic rapide à partir des diverses fonctions qu'ils proposent.

La solution présentée permet une interprétation rapide et efficace, à un très haut niveau, des paramètres extraits de l'analyse en temps réel des signaux d'origine physiologique.

Les six points suivants présentent les éléments qui caractérisent l'originalité d'un tel système.

1- C'est un matériel de chevet compact.

L'intégration des différentes fonctions de traitement et d'analyse des signaux sur le site même des interventions, permet au praticien de disposer, sur un seul appareil de dimensions modestes, de l'ensemble des informations qui peuvent lui apporter une aide dans son travail.

2- Les différentes unités de traitement sont modulaires et autonomes.

La modularité du système est effective, aussi bien pour le matériel que pour le logiciel. Ceci signifie, en particulier, que toute modification, ou toute adjonction au système, ne remet pas en cause la structure de ce qui existe, et ne perturbe en rien l'ensemble du système. De même, toute amélioration relative à un module, peut être réalisée aisément, sans contrainte sur l'environnement.

3- La mémorisation statistique du passé physiologique s'étend sur 24 heures.

De cette façon, il est possible de visualiser tous les paramètres enregistrés dans une tranche de temps donnée.

4- Il est possible de disposer de différents modes de représentation:

- Représentation d'un signal avec possibilité de gel à un moment donné, mais aussi et surtout, effet de zoom sur la trace.

- Représentation sous forme synthétique de tous les résultats (Histogrammes).

- Représentation temporelle ou spatiale des situations permettant la visualisation des corrélations existant entre les différents signaux observés.

5- On peut envisager le traitement intégré d'autres signaux.

La modularité rappelée ci-dessus, offre la possibilité de prévoir l'évolution de cet appareil par l'adjonction de cartes de traitement relatives à des signaux de nature différente, provenant d'autres services (neurologie, cardiologie, traumatologie...), ainsi que les algorithmes de décision correspondants.

6- C'est un élément important pour l'aide apportée à la décision dans l'acte médical.

L'évaluation précise des tendances physiologiques d'un patient, en fonction des situations rencontrées et analysées, peut donner des indications relativement précises sur sa pathologie.

Tous ces éléments contribuent à la constitution d'un appareil compact, intégré, évolutif, susceptible, par ses capacités de mémorisation et de représentation des événements, d'apporter une aide efficace au diagnostic médical.

## BIBLIOGRAPHIE

- /BERT/ BERTOYE P.H., DITTMAR A., DELHOMME G., MARICHY J.,  
UNTERREINER R.  
"Les grands brûlés: surveillance assistée par  
ordinateur et régulation de l'état thermovasculaire  
et métabolique."  
- Innov.Tech.Biol.Med., Vol 6, N° 3, 1985, P 322-340.
- /BERI/ BERTIN J.  
"Représentation graphique."  
- Encyclopaedia Universalis, VOL 7, 1979.
- /BLON/ BLONDEAU M. HILTGEN M.  
"Electrocardiographie clinique."  
- MASSON Editeurs, 120 Boulevard Saint-Germain  
75006 PARIS.
- /CAL1/ CALVEZ J.P.  
"Une méthodologie de conception des multi-micro-  
ordinateurs pour les applications en temps réel."  
- Thèse d'Etat - ENSM - Nantes, Novembre 1982.
- /CAL2/ CALVEZ J.P., THOMAS Y.  
"Méthodologie de conception des multi-micro-  
ordinateurs de commande des processus complexes."  
- Rapport final ATP N° 9.81.63, Nantes, Mai 1983.
- /CHAM/ CHAMBRIN M.C.  
"Réalisation à l'aide de microprocesseurs d'un  
système de surveillance de l'efficacité  
ventilatoire en réanimation respiratoire."  
- Thèse de docteur-ingénieur, U.S.T.L.,  
Décembre 1979.

/CHEB/ CHEBEL B.

"Application à l'analyse de structure et de la fonction entropie au traitement du signal."

- Thèse de docteur-ingénieur, U.S.T.L., Mars 1983.

/CONG/ CONGRES IRIES

"Journées d'informatique médicale de Toulouse 1982. Association pour l'informatique et la recherche en informatique, électronique et statistique."

- B.P. 14, 78650 BEYNES.

/CROC/ CROCUS

"Systèmes d'exploitation des ordinateurs."

- Dunod 1975.

/DIJK/ DIJKSTRA E.W.

"Hierarchical ordering of sequential processes."

- Acta Informatica, Vol. 1,2, 1971, P. 115-138

/DOMI/ DOMINE F., DUPONT D., MEHADJI M., CHAMBRIN M.C.,  
COUVREUR M., CHOPIN C.

"Conception d'un appareillage intégré de surveillance individuelle des malades en réanimation."

- Innov.Tech.Biol.Med., Vol 6, N° 4, 1985, P 373-380.

/DUPO/ DUPONT D.

"Observation et aide à la décision: application à la conception d'un observateur numérique couplé à un système expert."

- Thèse 3ème cycle, U.S.T.L., Janvier 1986.

/EGIC/ LABORATOIRE EGIC

"Réanimation médicale et chirurgicale. Tome 2."

- Laboratoire Egic, B.P. 172, 45203 MONTARGIS CEDEX.

/FATH/ FATHI E.T, KRIEGER M.

"Multiple microprocessor systems: What, Why, When."

-I.E.E.E. Computer, Mars 1983, P 23-32.

- /FONT/ FONTE C.  
"Multiprocesseurs temps réel."  
- Mémoire de D.E.A., U.S.T.L., Juin 1982.
- /GAUD/ DE GAUDEMARIS R., VILLEMMAIN P., DAVER H.,  
DIMITRIOU R., MALLION J.M.  
"Système d'acquisition et de traitement de mesures  
automatiques de la pression artérielle réalisées en  
ambulatoire. Aide au diagnostic médical."  
- Innov. Tech. Biol. Med., Vol 5, N° 2,  
1984, P 189-198.
- /GEOR/ GEORGES R., SAUMON G., BARRET G., TURIOF J.  
"Traitement numérique automatique de signaux  
analogiques. Application à la physiopathologie  
respiratoire."  
-Ann. Med. Interne, Janvier 1972, P. 43-49.
- /GOTH/ GOTHARD J.W.W., BUSST C.M., BRANTHWAITE M.A.,  
DAVIES N.J.H., DENISON D.M.  
"Applications of respiratory mass spectrometry to  
intensive care."  
- Anaesthesia, 1980, Volume 35, P. 890-895.
- /HEW1/ HEWLETT PACKARD  
"Moniteur de patient 78353."  
- Hewlett Packard France 1983.
- /HEW2/ HEWLETT PACKARD  
"An algorithm for reduction of respiration  
artefact in pulmonary artery pressure measurement."  
- Hewlett Packard U.S.A. 1983.
- /HEW3/ HEWLETT PACKARD  
"Self-test and serviceability for dependable  
central patient monitoring."  
- Hewlett Packard journal, November 1980.
- /HEW4/ HEWLETT PACKARD  
"Plain talk about computerized arhythmia  
monitoring."  
- Hewlett Packard U.S.A. 1984.

- /ITOH/ ITOH A., ISADA A., KUCHI N.K.  
"Microcomputer-based respiratory function  
monitoring system using impedance pneumographie."  
- Med. & Biol. Eng. & Comput. September 1982,  
P. 620-625.
- /INT1/ iSBC 80/24 single board computer hardware reference  
manual. Manual order number: 142648-002.  
- Intel Corporation, 3065 Bowers Avenue, Santa Clara  
CA 95051.
- /INT2/ SBC 80/05 single board computer hardware reference  
manual. Manual order number: 9800483A.  
- Intel Corporation, 3065 Bowers Avenue, Santa Clara  
CA 95051.
- /INT3/ iSBC 254 single board computer hardware reference  
manual. Manual order number: 113844.  
- Intel Corporation, 3065 Bowers Avenue, Santa Clara  
CA 95051.
- /INT4/ iSBC 660 system chassis hardware reference  
manual. Manual order number: 9800505-04.  
- Intel Corporation, 3065 Bowers Avenue, Santa Clara  
CA 95051.
- /INT5/ iSBC power supply hardware reference  
manual. Manual order number: 9800803-03.  
- Intel Corporation, 3065 Bowers Avenue, Santa Clara  
CA 95051.
- /INT6/ iSBC 016A/032A/064A/028A/056A ram board hardware  
reference  
manual. Manual order number: 143572-001.  
- Intel Corporation, 3065 Bowers Avenue, Santa Clara  
CA 95051.
- /KRAK/ KRAKOWIAK S.  
"Principes des systèmes d'exploitation des  
ordinateurs."  
- Dunod 1985.

- /LABA/ LABARERE M., KRIEK J.P., GIMONET B.  
"Le filtrage et ses applications."  
- Cepadues-éditions, Toulouse.
- /LACO/ LACOSTE J., HANNHART D.  
"Composantes circulatoires et ventilatoires de la ductance du CO<sub>2</sub>."  
- Ann. Med. Nancy, 1972, 11, P 277-286.
- /LECE/ LECERF J.M.  
"Conception d'une structure logicielle redondante en vue d'améliorer la sureté des prises de décision en temps réel: application à l'analyse automatique des signaux biologiques."  
- Thèse 3ème cycle, U.S.T.L., Octobre 1983.
- /LEDL/ LEDLEY R.S.  
"Practical problems in the use of computers in medical diagnosis."  
- Proceeding of the I.E.E.E., Vol.57, N° 11, November 1969, P. 1900-1918.
- /LIU/ LIU C.L., LAYLAND J.W.  
"Scheduling algoritms for multi-programming in a hard-real-time environnement."  
- J.A.C.M., Vol 26, Janvier 1973.
- /LHER/ LHERMITTE CL.  
"Les systèmes d'exploitation: structure et concept fondamentaux."  
- Masson 1985.
- /LIST/ LISTER A.M.  
"Principes fondamentaux des systèmes d'exploitation."  
- Eyrolles 1985

/MEHA/ MEHADJI M.

"Le monitoring cardiorespiratoire: conception et réalisation d'un système d'exploitation."

- Thèse de docteur-ingénieur, Université du Droit et de la Santé de Lille II, Faculté de Pharmacie, Février 1986.

/MERI/ MERIAUX M.

"Contribution à l'imagerie informatique. Aspects algorithmiques et architecturaux."

- Thèse d'état, Lille 1, Juillet 1984.

/MURA/ MURAT K.

"Traitement numérique des signaux."

- Editions Georgie.

/OSBO/ OSBORN J., BEAUMONT J.O., RAISON J.C.A.,  
RUSSELL J., GERBODE F.

"Measurement and monitoring of acutely ill patient by digital computer."

- Surgery, December 1968, P 1057-1070.

/PELE/ PELERIN M.

"Synthèse d'images et parallélisme: algorithmes et architecture."

- Thèse 3ème cycle, Lille 1, Janvier 1985.

/RAJA/ RAJACOPALAN C.V.

"Segmentation des suites temporelles: application à l'extraction heuristique des paramètres fondamentaux du cardiorespirogramme en pathologie néonatale."

- Thèse de docteur-ingénieur, Lille 1, 1979.

/ROUC/ ROUCAIROL R.

"représentation du parallélisme et modèles de systèmes."

- Bulletin de liaison de la recherche en informatique, N° 83, 1983.

- /THOR/ THORAVAL R.  
"Un outil informatique pour l'analyse graphique des données."  
- Irisa publications N° 215, Juin 1983.
- /TREM/ TREMOLIERES J.  
"La mesure de la compliance et des résistances pulmonaires."  
- Electronique applications N° 6, P. 51-58.
- /VAS1/ VASSEUR C., CHEBEL B., COUVREUR M., TOULOTTE J.M.,  
DUBOIS O.  
"Signaux physiologiques et micro-informatique:  
Exemple de réalisation d'une instrumentation  
intégrée."  
- Innov.Tech.Biol.Med. Vol 1, N° 5, 1980,  
P. 92-104.
- /VAS2/ VASSEUR C., RAJAGOPALAN C., COUVREUR M.,  
TOULOTTE J.M., DUBOIS O.  
"A microprocessor oriented segmentation technique:  
an efficient tool for electrophysiological signals  
analysis."  
- I.E.E.E. Trans. on instrumentation and  
measurement, IM-28, N° 4, December 1979, P. 259-262.
- /VAS3/ VASSEUR C.  
"La notion d'évènement dans les systèmes dynamiques:  
détection, classification temps réel et application à  
la conception d'une instrumentation distribuée."  
- Thèse d'état - Université des sciences et techniques  
de Lille, juin 1982.
- /VERJ/ VERJUS J.P.  
"Parallélisme et synchronisation."  
- Bulletin de liaison et de recherche en informatique  
et automatique, N° 83, 1983.
- /VICT/ VICTOR J., GESLIN P., TADET A.  
"Système modulaire d'analyse des arythmies  
cardiaques."  
- Biosygma 1978, Communication A.III.5., P. 216-220.

/WIES/ WIESS S.M., KULIKOWSKI C.A., AMAREL S., SAFIR A.  
"A model-based method for computer-aided medical  
decision-making."  
- Artificial intelligence 11(1978), North-Holland  
Publishing Company, 1978, P. 145-171.

/WILL/ WILLIAMS B.T., HOLL J.E., ELSTON J.  
"Pattern interpretation in laboratory data via  
computer graphics."  
- Biosigma 1978, Tome 1, Paris 24-28 Avril 1978,  
P. 490-495.

