

80/23

UNIVERSITÉ DE NANCY 1

Sc N 80 / 67 B

THESE

PRÉSENTÉE A

L'UNIVERSITÉ DE NANCY 1

POUR OBTENIR LE TITRE DE

DOCTEUR - INGENIEUR EN AUTOMATIQUE

PAR

Jean-Pierre DRAPIER-BÊCHE



. DÉFINITION ET RÉALISATION D'UN CALCULATEUR MULTIPROCESSEURS

ADAPTÉ A LA COMMANDE EN TEMPS RÉEL DE PROCESSUS MULTITACHES.

. APPLICATION A UNE MACHINE DE DÉCOUPE OPTIMALE DE PLANCHES.

Soutenue publiquement le 24 JUIN 1980 devant la commission d'examen.

Membres du jury:

Président : Monsieur M. AUBRUN

Examineurs: Messieurs J. FEUTRIER

A. FRUHLING

R. HUSSON

J.-P. MUSSE

J. PECHAUD

M. VERON

BIBLIOTHEQUE SCIENCES NANCY 1



D 095 180157 2

UNIVERSITÉ DE NANCY I

THESE

PRÉSENTÉE A

L'UNIVERSITÉ DE NANCY I

POUR OBTENIR LE TITRE DE

DOCTEUR - INGENIEUR EN AUTOMATIQUE

PAR

Jean-Pierre DRAPIER-BÊCHE



. DÉFINITION ET RÉALISATION D'UN CALCULATEUR MULTIPROCESSEURS

ADAPTÉ A LA COMMANDE EN TEMPS RÉEL DE PROCESSUS MULTITACHES.

. APPLICATION A UNE MACHINE DE DÉCOUPE OPTIMALE DE PLANCHES.

Soutenue publiquement le 24 JUIN 1980 devant la commission d'examen.

Membres du jury:

Président : Monsieur M. AUBRUN

Examineurs: Messieurs J. FEUTRIER

A. FRUHLING

R. HUSSON

J.-P. MUSSE

J. PECHAUD

M. VERON

- AVANT - PROPOS -

Je tiens à exprimer ma sincère reconnaissance à Monsieur le Professeur M. AUBRUN pour la bienveillance qu'il m'a toujours témoignée et pour l'honneur qu'il me fait en présidant la Commission d'Examen.

Je remercie Monsieur J. FEUTRIER de l'intérêt qu'il porte à mon travail et de sa participation à mon jury.

Que Monsieur le Professeur A. FRUHLING, Directeur du Laboratoire d'Electricité et d'Automatique, trouve ici l'expression de ma profonde gratitude. Je le remercie de l'estime qu'il me manifeste en participant à mon jury.

Je suis très sensible de l'honneur que me fait le Professeur R. HUSSON en participant à la Commission d'Examen.

Je remercie tout particulièrement Monsieur Jean-Pierre MUSSE, avec qui j'ai réalisé le système de découpe optimale de planches, pour les conseils qu'il m'a donnés et d'avoir bien voulu examiner dans le détail mon mémoire.

Monsieur J. PECHAUD est à l'origine du contrat d'étude du calculateur multiprocesseurs. Qu'il trouve ici l'expression de ma reconnaissance pour l'intérêt qu'il a porté à mon travail et pour sa participation à mon jury.

Ce travail a été effectué dans le cadre des activités du Centre Universitaire de Commande Numérique qu'anime avec dynamisme Monsieur le Professeur VERON. Je tiens à le remercier vivement pour ses encouragements constants et pour l'ambiance amicale qu'il sait faire régner dans son groupe.

Mes camarades du Laboratoire et en particulier ceux du groupe
Commande Numérique m'ont apporté une collaboration constructive et
sympathique. Qu'ils trouvent ici l'expression de mes remerciements
amicaux.

Je remercie Mademoiselle A. KASTNER pour la qualité et la
rapidité de son travail dans la réalisation matérielle de ce mémoire.

Enfin, je tiens à remercier mon épouse pour la compréhension
et le soutien moral qu'elle n'a cessé de me prodiguer tout au long
de ce travail.

A Colette,
 Nathalie,
 Stéphane,
 Sabrina,

à mes parents.

TABLE DES MATIERES

Première Partie : DEFINITION ET REALISATION D'UN CALCULATEUR MULTIPROCESSEURS

INTRODUCTION

1

CHAPITRE I : DEFINITION DU CALCULATEUR MULTIPROCESSEURS

I-1	Situation du calculateur dans une structure hiérarchisée	3
I-2	Intérêt d'un calculateur multiprocesseurs	6
I-3	Tâches confiées à un calculateur	7
I-4	Principes de base de synchronisation et de communication	9
I-5	Modalités de transition et de dialogue	11

CHAPITRE II : REALISATION DU CALCULATEUR MULTIPROCESSEURS

II-1	Introduction	15
II-2	Choix du microprocesseur	15
II-3	Présentation générale du calculateur	16
II-4	Présentation des différents éléments du calculateur	18
II-4- 1	Bus locaux et communs	18
II-4- 2	Signaux de service locaux	20
II-4- 3	Signal de service commun	22
II-4- 4	Horloge	22
II-4- 5	Décodage des adresses mémoire	22
II-4- 6	Décodage des adresses des registres d'entrée-sortie	24
II-4- 7	Mémoire locale	25
II-4- 8	Eléments de dialogue	25
II-4- 9	Système d'interruption	27
II-4-10	Mise en attente du processeur	32
II-4-11	Mot d'état du microcalculateur	32
II-5	Insertion du calculateur dans une structure hiérarchisée	32

.../...

II-6	Mémoire commune	33
II-6-1	Allocateur	33
II-6-2	Type de mémoire utilisée	35
II-6-3	Temps d'attente des données	35
II-6-4	Interchangeabilité des microprocesseurs	36

CHAPITRE III : SYSTEME DE MISE AU POINT ET DE MAINTENANCE

III-1	Introduction	37
III-2	Version prototype	37
III-2-1	Signaux agissant sur les microprocesseurs contrôlés	40
III-2-2	Action matérielle sur le fonctionnement du microprocesseur	41
III-2-3	Action logicielle sur le fonctionnement du microprocesseur	41
III-3	Test et maintenance du calculateur	42
III-4	Dépannage du processus industriel	43

Deuxième partie : APPLICATION A UNE MACHINE DE DECOUPE OPTIMALE DE PLANCHES

<u>INTRODUCTION</u>	45
---------------------	----

CHAPITRE I : CAHIER DES CHARGES ET INTERFACES DE COUPLAGE

I-1	Introduction	46
I-2	Entrée des listes de découpes	46
I-3	Mesure d'un produit	47
I-3-1	Détermination des longueurs utiles et de la qualité	47
I-3-2	Section et qualité d'un produit	49
I-3-3	Mesure de la planche	50
I-4	Découpe des planches	50
I-5	Asservissement en position de la planche	53
I-6	Cartes de couplage calculateur-machine	55

CHAPITRE II : ORGANISATION DU LOGICIEL

II-1	Introduction	57
II-2	Organisation de la mémoire commune	59
II-3	Tâche DIALOGUE	62
II-3-1	Commandes codées	65
II-3-2	Interpréteur de commande	66

II-4	Tâche MESURE	68
II-5	Tâche OPTIMISATION	69
	II-5-1 Programme d'optimisation	70
II-6	Tâche DECOUPE	72
	II-6-1 Programme de découpe	75
II-7	MONITEUR	76
<u>CONCLUSION</u>		77

- 1ÈRE PARTIE -

DEFINITION ET REALISATION D'UN CALCULATEUR

MULTIPROCESSEURS ADAPTE A LA COMMANDE

EN TEMPS REEL DE PROCESSUS MULTITACHES

I N T R O D U C T I O N

L'apparition du minicalcateur, et plus récemment celle du microprocesseur, a bouleversé la commande des systèmes industriels. Dans le domaine de la Commande Numérique de Machines Outils, la commande par calcateur a déjà fait ses preuves et l'utilisation de l'ordinateur pour la gestion des ateliers de fabrication mécanique est chose irréversible.

Mais il existe de nombreuses machines spéciales, fabriquées par de petites sociétés, qui gagneraient en performance si elles étaient pilotées par un calcateur. Si ces sociétés veulent franchir le pas de la commande numérique, elles ont le choix entre le minicalcateur industriel et les cartes de développement proposées par les fabricants de microprocesseurs.

Le calcateur industriel est difficilement modulable en puissance et ne s'adapte pas toujours aux besoins réels d'une application, surtout si cette application est susceptible d'évoluer ou d'être vendue en version plus ou moins sophistiquée.

La carte de développement uniprocasseur, de moindre coût, n'est pas toujours suffisante pour répondre aux exigences du processus et les extensions possibles sont très vite limitées.

Par contre, un calcateur multimicroprocesseurs dont la puissance serait adaptée à l'application et où chaque tâche serait confiée à un microprocesseur, constitue une solution séduisante.

Cette possibilité a été vite pressentie puisque la Société R2E commercialise depuis 1975 le MICRAL-M construit par association de microordinateurs utilisant un microprocesseur 8080 de la Société INTEL. Depuis, certains fabricants de microprocesseurs proposent des cartes de développement connectables à un bus commun (6).

Mais ces cartes offrent des possibilités souvent inutilisées, ce qui ne permet pas d'adapter au mieux le matériel au problème à traiter et alourdit le prix de l'ensemble.

Le travail que nous présentons dans ce mémoire est le résultat d'une recherche de définition d'un microcalculateur minimal capable de piloter une tâche en temps réel ; la mise en parallèle de plusieurs modules devant permettre la réalisation d'un calculateur multiprocesseurs.

Notre préoccupation essentielle a été de concevoir un système simple avec un minimum de matériel afin qu'il puisse être éventuellement réalisé par les utilisateurs.

Il est donc nécessaire de proposer conjointement un système permettant de tester le bon fonctionnement des cartes après fabrication.

D'autre part, pour la mise en oeuvre du logiciel, nous avons réalisé un système de mise au point adapté à ce type de calculateur et qui permette d'avoir accès aux différents processeurs.

La présentation du système multiprocesseur et de ses outils de développement fait l'objet de la première partie de ce mémoire. La deuxième partie est consacrée à une application qui nous a été commandée par un industriel du bois, et qui nous a permis de vérifier la validité de notre calculateur.

L'analyse détaillée du calculateur multitâche et surtout du logiciel de mise en oeuvre pour la conduite de la machine demanderait de longues explications ne présentant qu'un intérêt limité.

Nous avons donc été amenés à simplifier certaines descriptions afin d'apporter de la clarté au texte et à nous consacrer à l'essentiel.

Ce travail important, mené à bien dans les délais impartis, a été réalisé avec la collaboration très efficace de Jean-Pierre MUSSE qui s'est plus particulièrement occupé du logiciel de commande de la machine.

CHAPITRE I

DEFINITION DU CALCULATEUR MULTIPROCESSEURS

I-1) SITUATION DU CALCULATEUR DANS UNE STRUCTURE HIERARCHISEE

L'approche actuelle de la commande d'une unité industrielle complexe se fonde sur les concepts de commande décentralisée, d'acquisition répartie et de traitement distribué des données.

La mise en oeuvre de ces concepts fait appel à une hiérarchie d'ordinateurs, accomplissant chacun des tâches appropriées à sa situation dans cette hiérarchie (1).

L'élément supérieur de la pyramide (*figure I.1.*), affecté à l'organisation générale de la production, définit les objectifs globaux en fonction des paramètres de planification, des contraintes humaines et économiques et des possibilités de production. Les ordres de production sont envoyés aux divers calculateurs régionaux implantés près des lieux de fabrication. Ceux-ci, en retour, rendent compte de leur production permettant ainsi à l'ordinateur fédéral d'éditer les résultats, de tenir à jour les stocks, de signaler les arrêts et éventuellement, d'élaborer une nouvelle stratégie.

Les calculateurs régionaux organisent le travail journalier d'un ensemble d'unités de production en tenant compte des contraintes et des objectifs locaux. Ils connaissent constamment l'état et la production de chaque machine afin de pouvoir réagir sur les éléments amont et aval de la chaîne de production. Un couplage est nécessaire entre calculateurs régionaux lorsque ceux-ci travaillent sur un même produit et qu'une coordination est indispensable.

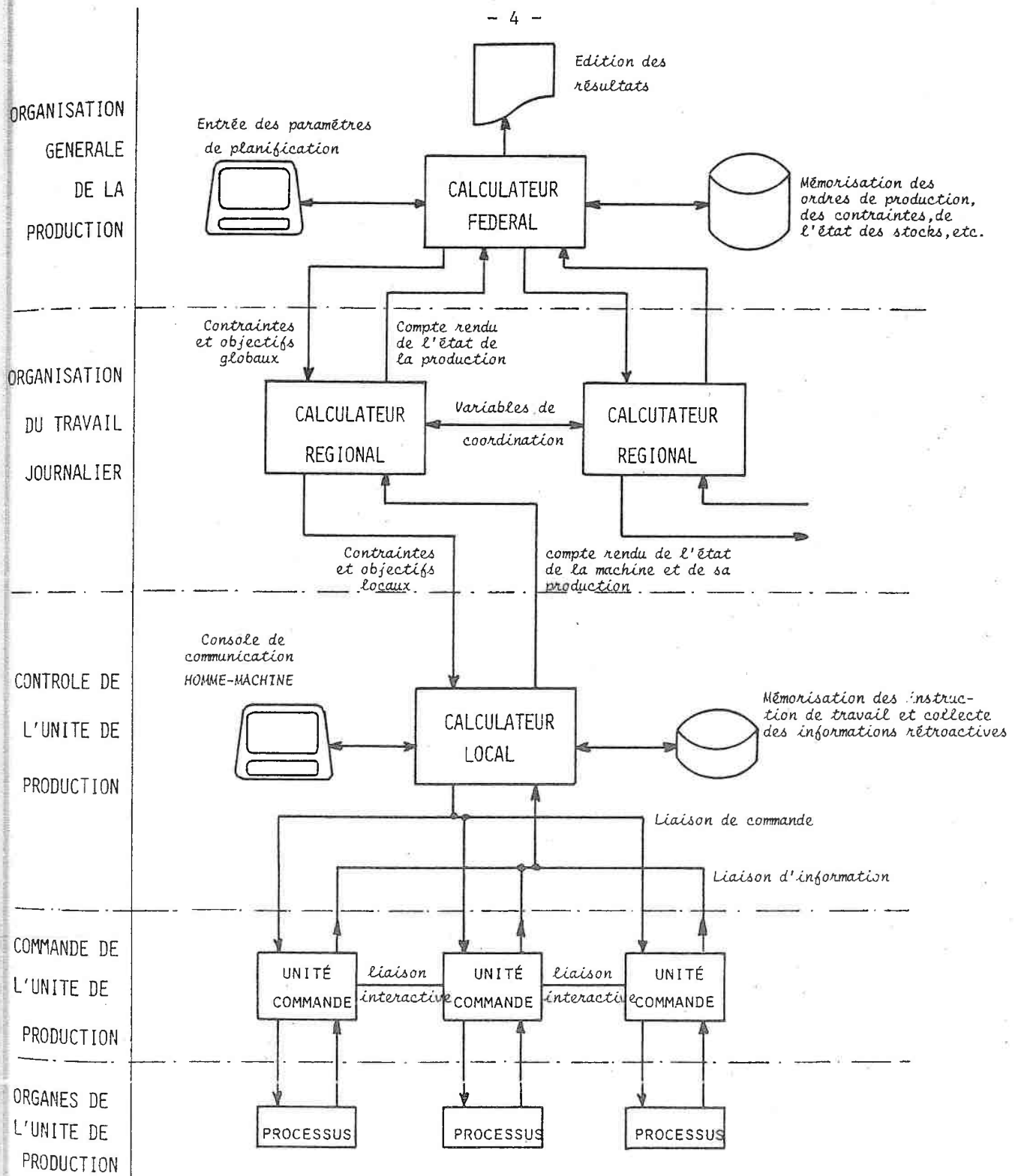


Figure I.1- Structure hiérarchisée d'ordinateurs.

Enfin, l'unité de production est composée de plusieurs processus contrôlés chacun par une unité de commande qui élabore les ordres relatifs à la tâche à effectuer. Les diverses unités de commande disposent entre elles de liaisons interactives et sont sous le contrôle d'un calculateur local qui assure l'activation et la synchronisation des différentes tâches.

Le calculateur que nous avons étudié se situe à ce niveau de la pyramide et est principalement destiné au contrôle et à la commande de petites unités de production.

Les liaisons interactives qui permettent l'échange de données et la synchronisation des différentes tâches peuvent être de deux types :

- liaisons biprocesseurs qui autorisent seulement l'échange entre deux unités de commande (*figure I.2.*). Les transits des informations et des ordres sont alors effectués en cascade,

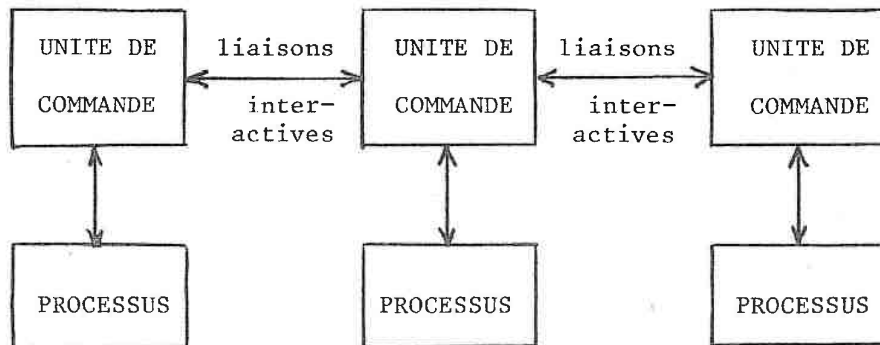


Figure I.2- Liaisons biprocesseurs.

- liaisons multiprocesseurs qui permettent à tous les éléments de communiquer entre eux (*figure I.3.*).

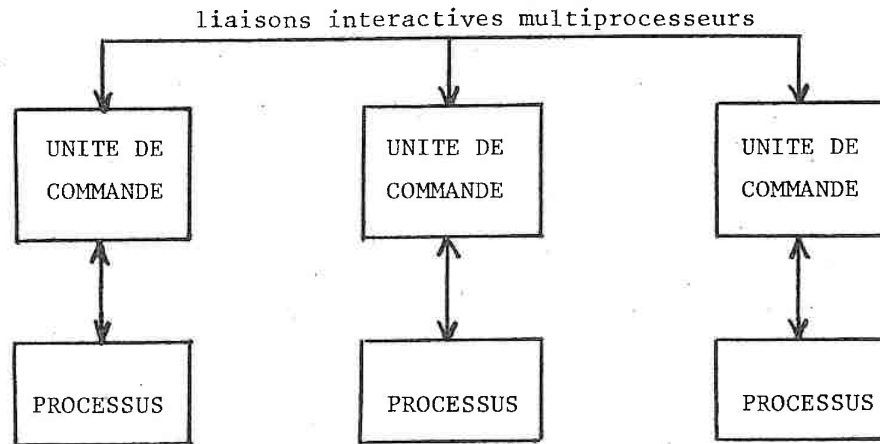


Figure I.3- Liaisons multiprocesseurs

C'est ce type de liaison qui a été retenu pour notre étude, car c'est lui qui apporte le plus de souplesse d'utilisation.

Les communications entre les différents calculateurs d'une structure hiérarchisée sont assurées par des modes de transmission adaptés à chaque niveau de la hiérarchie.

Le transfert d'information entre deux calculateurs dépend essentiellement de la vitesse d'échange désirée et de leur éloignement.

Pour des distances supérieures à 5 m, on utilise généralement une transmission série. Ce canal peut consister en une simple paire de fil torsadé, un câble coaxial ou une liaison par fibre optique lorsque l'environnement est fortement bruité. Pour les transferts rapides et proches, la transmission parallèle est la mieux adaptée.

Etant donné que notre ordinateur est destiné à de petites unités de production dont les organes à commander sont relativement groupés, nous avons décidé de rassembler les unités de commandes et d'utiliser un bus parallèle pour les transferts interprocesseurs.

I-2) INTERET D'UN CALCULATEUR MULTIPROCESSEURS

L'unité de commande met en oeuvre généralement un mini-calculateur qui déroule un programme séquentiel. Or en pratique, de nombreux systèmes industriels exécutent plusieurs tâches en parallèle, ce qui

pose le problème de la hiérarchisation des traitements par le calculateur. D'autre part, une tâche très prioritaire et longue peut accaparer le calculateur et ne plus laisser assez de temps aux autres tâches.

Actuellement, l'utilisation de nouveaux composants électroniques hautement intégrés, tels que les microprocesseurs, autorise la construction de microcalculateurs de bas prix qui, mis en parallèle, peuvent constituer un calculateur multiprocesseurs ayant de meilleures performances et dont la qualité essentielle est la modularité.

Cette modularité est double :

- modularité matérielle qui assure une grande flexibilité de mise en oeuvre, permettant une meilleure adaptation aux processus,
- modularité logicielle qui autorise l'écriture des différentes tâches de façon pratiquement indépendante et permet l'adjonction de nouveaux modules sans modification importante du logiciel existant.

D'autre part, la modularité matérielle assure une interchangeabilité et une maintenance aisée et rapide.

I-3) TACHES CONFIEES A UN CALCULATEUR

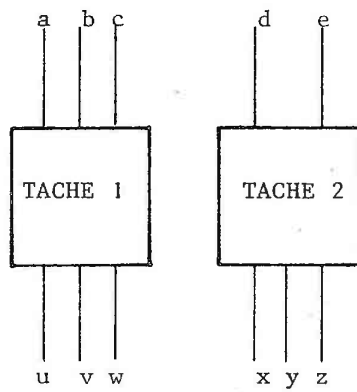
Les tâches confiées à un calculateur dans la commande d'un processus sont généralement de 2 types :

- des tâches d'entrées-sorties,
- des tâches d'élaboration de grandeurs de commandes.

Ces tâches peuvent être indépendantes, parallèlement ou séquentiellement coopérantes (3) (figure I.4.). Ceci implique de fixer le niveau de priorité de traitement des différentes tâches, de lever les conflits d'utilisation des ressources communes et de définir le moyen de communication entre tâches.

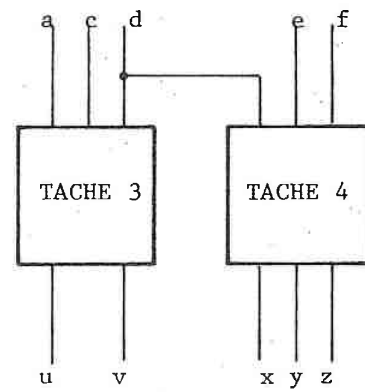
Pour résoudre ces problèmes, la mise en oeuvre d'un ordinateur classique implique une organisation du logiciel autour d'un moniteur qui assure la répartition des travaux dans le temps pour éviter d'éventuels conflits.

Cette solution n'est pas idéale lorsque les diverses tâches du processus doivent être effectuées en simultanéité vraie, à cause de leur fréquence d'apparition ou de la rapidité d'exécution qu'elles nécessitent.



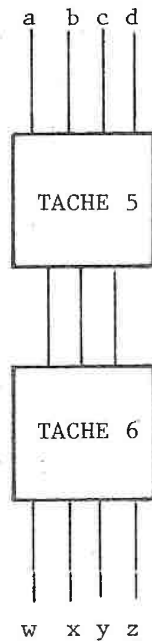
TACHES INDEPENDANTES

Execution asynchrone



TACHES PARALLELEMENT COOPERANTES

dont l'execution est synchronisée
par la variable d.



TACHES SEQUENTIELLEMENT COOPERANTES

Le lancement de la tâche 6 est
conditionné par la fin de la tâche 5.

Figure I.4-

Par contre, une structure multiprocesseurs où chaque processeur contrôle une tâche particulière permet un véritable traitement en parallèle. Les différentes tâches du processus à commander déterminent le nombre de processeurs nécessaires entre lesquels il faut définir des moyens de synchronisation et de communication.

I-4) PRINCIPES DE BASE DE SYNCHRONISATION ET DE COMMUNICATION

L'analyse des diverses tâches à accomplir dans la conduite de la plupart des processus montre que l'on peut se limiter à des procédures de communication et de synchronisation simples. Ceci nous a amené à définir les principes de bases suivants :

- plusieurs tâches peuvent être confiées à un même module si aucun conflit d'exécution n'est possible, c'est-à-dire, si elles s'excluent mutuellement,
- les tâches, généralement séquentiellement coopérantes, communiquent par l'intermédiaire d'une mémoire "boîte à lettre" accessible par toutes les tâches,
- cette mémoire constitue la seule ressource commune. Ainsi un périphérique commun à plusieurs tâches doit être géré par un processeur spécialisé qui règle les conflits éventuels d'utilisation de ce périphérique,
- pour ne pas nuire au fonctionnement de l'ensemble, il est interdit d'utiliser dans une tâche des boucles d'attente avec tests de drapeaux ou sémaphores situés dans la mémoire commune,
- la coordination des tâches est exclusivement assurée par un module "moniteur" qui dispose de rapports privilégiés avec les autres processeurs,
- les interruptions propres à chaque tâche sont connectées sur le module correspondant, ce qui minimise le problème de leur hiérarchisation,
- l'affectation dynamique des tâches aux divers modules ne se justifie pas pour les utilisations envisagées. Le travail d'affectation est donc laissé au concepteur qui en réalise la partition, de façon à utiliser au mieux chaque module du système.

Ces principes nous conduisent à la structure multiprocesseurs de la figure I.5. Le module "MONITEUR" et les modules "ESCLAVE" ont accès

à la mémoire "BOITE A LETTRE" en utilisant un bus commun. Les entrées-sorties sont connectées aux bus locaux des modules "ESCLAVE" concernés. Les interruptions extérieures et les signaux de dialogue arrivent directement sur chaque carte "ESCLAVE".

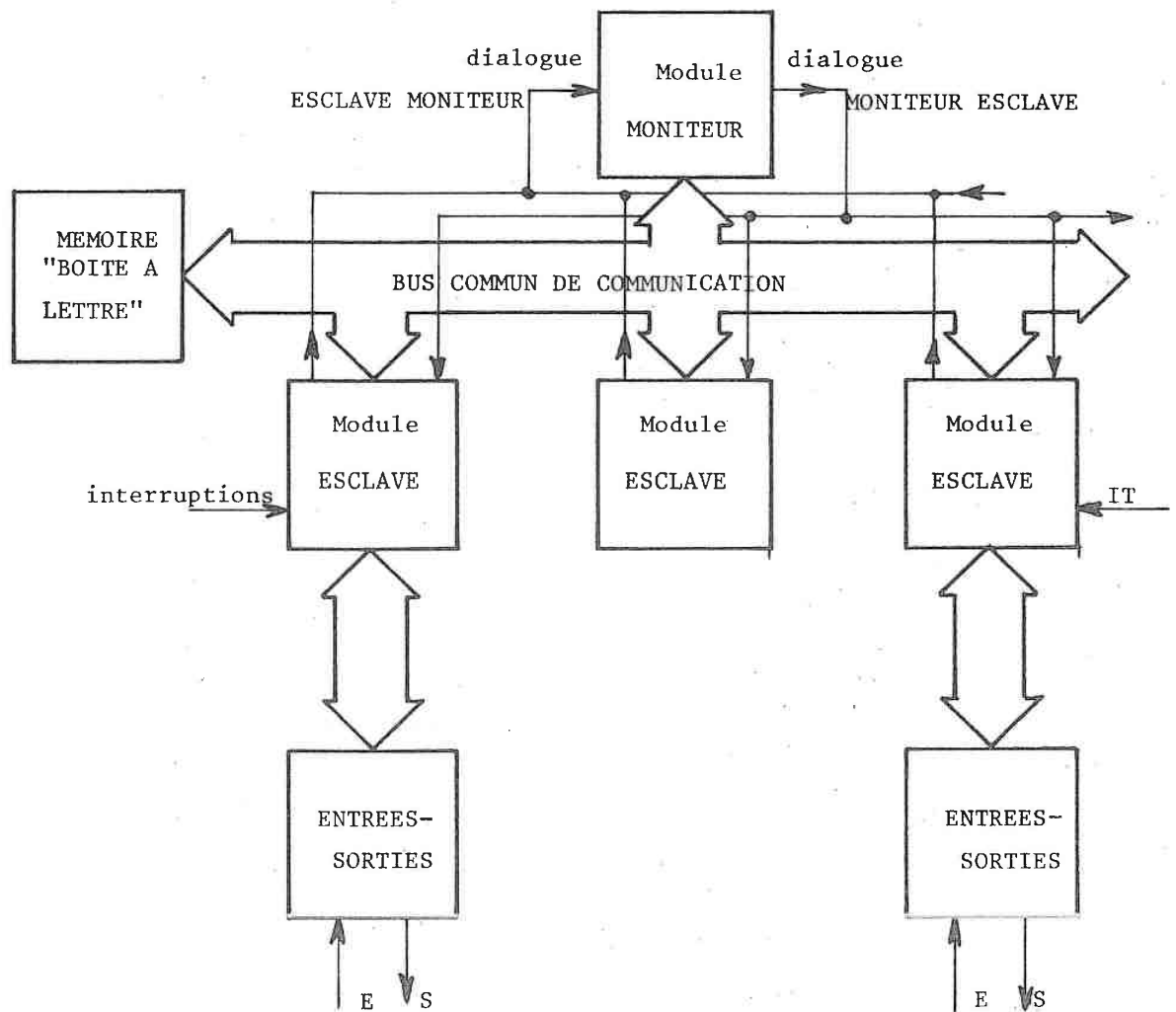


Figure I.5- Structure multiprocesseur retenue

I-5) MODALITES DE TRANSITION ET DE DIALOGUE

Le déroulement temporel de chaque tâche est à tout instant repéré par son état. Quatre états distincts, observables par le module MONITEUR, sont définis :

- PRET : il correspond à l'état initial dans lequel doit se trouver toute tâche avant une première activation, et dans lequel il faut donc la remettre lors d'un arrêt global du système. Cet état dépend des exigences de sécurité nécessaires à la mise sous tension,

- EN ATTENTE : la tâche se met elle-même en attente, soit en fin d'exécution, soit après avoir demandé des informations en provenance d'une autre tâche, soit en attente d'un évènement dépendant d'un autre module,

- SUSPENDU : par décision du module MONITEUR, une tâche peut être momentanément ou définitivement arrêtée. Elle ne peut quitter cet état que sur une autre intervention du module MONITEUR,

- ACTIF : il correspond à la phase d'exécution proprement dite qui peut comporter des attentes de ressources propres, chaque tâche étant chargée de gérer ses interruptions externes propres.

Une transition correspond au passage d'une tâche d'un état dans un autre. Elle peut être commandée par le module MONITEUR ou issue du module ESCLAVE support de la tâche concernée.

Le dialogue inhérent à ces transitions, est effectué par l'intermédiaire de la mémoire commune. Elle contient une liste des états correspondant aux fonctions réalisées par chaque module, et les indications nécessaires à l'identification, par le module MONITEUR, des demandes qui lui sont adressées.

Grâce à ces éléments de dialogue, le module MONITEUR peut assurer la coordination harmonieuse des diverses tâches.

La figure I-6 met en évidence les états d'une tâche et les diverses transitions envisagées.

La transition 1 est provoquée par le module moniteur qui est le seul directement activé par la mise sous tension. Cette transition correspond à la transmission au module concerné du signal RESET qui provoque un démarrage à l'adresse zéro où l'on a pris soin de programmer

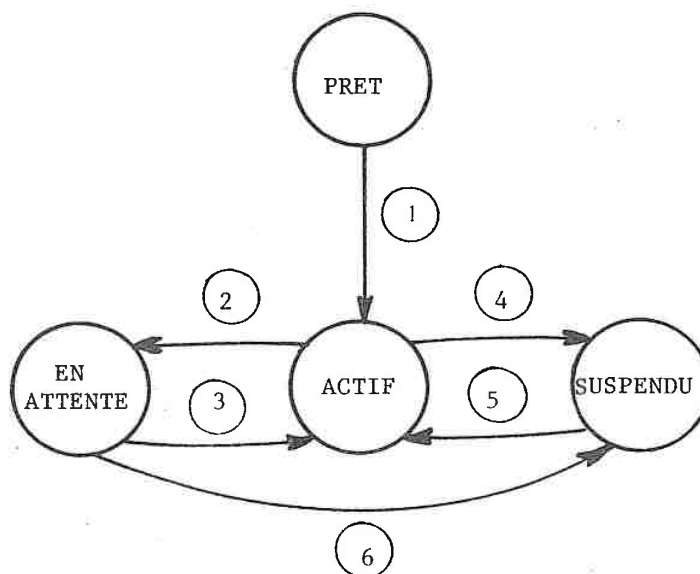


Figure I.6- Différents états d'une tâche

un branchement indirect à une place de la mémoire commune. Cette place, préalablement initialisée par le moniteur, permet donc au module MONITEUR de lancer un module ESCLAVE à une adresse déterminée.

La transition 2 est la seule qui puisse être provoquée depuis un module ESCLAVE qui génère une interruption vers le moniteur après avoir initialisé des indicateurs en mémoire commune. Ces indicateurs précisent au module MONITEUR le processeur qui a provoqué l'interruption, la nature de la transition effectuée et les modalités de réactivation de la tâche. La tâche entre alors EN ATTENTE par l'exécution de l'instruction HALT, qui provoque la mise en WAIT du processeur.

La transition 3 entraîne la reprise en séquence d'une tâche en attente. Pour cela, le moniteur génère une interruption qui provoque le retour à l'état RUN du processeur avec déroutement vers le traitement de l'interruption. Le retour du sous-programme d'interruption se fait en séquence de l'instruction HALT qui avait mis le processeur EN ATTENTE.

La transition 4 provoque la suspension de la tâche. Le moniteur met à zéro l'entrée PAUSE du processeur ESCLAVE qui passe alors à l'état WAIT. Le contexte de la tâche reste valide et son exécution peut être poursuivie ou abandonnée. Il faut noter que, bien que non traitées lorsque la tâche est suspendue, les interruptions éventuelles sont prises en compte si elles ne sont pas masquées.

Pour la transition 5, il faut envisager deux possibilités :

- soit la reprise en séquence de la tâche suspendue : pour ce faire, le module MONITEUR lève le signal PAUSE du module ESCLAVE concerné,
- soit l'abandon de la tâche suspendue et l'activation d'une autre tâche : le moniteur précise par un indicateur en mémoire commune la tâche à activer, génère une interruption et lève le PAUSE du module ESCLAVE. Le sous-programme d'interruption alors exécuté assure le branchement à l'adresse de début de la tâche à activer.

La transition 6 correspond à la suspension d'un module en attente lorsque l'exécution ne doit pas être poursuivie. L'action du moniteur est identique à celle de la transition 4.

Remarquons que nous n'avons pas retenu une transition pour mettre une tâche dans l'état PRET. En effet, il semble qu'une telle transition conduit à l'abandon de la tâche et de son contexte, et ne présente donc pas d'intérêt en régime permanent. Compte tenu de la définition de l'état PRET, cette transition est par contre nécessaire lorsqu'un arrêt du système est désiré.

Elle peut alors être provoquée de deux manières selon la nature de la tâche abandonnée :

- si, pour des raisons de sécurité la tâche nécessite une action préalable sur son environnement, il faut décrire le cycle SUSPENDU, ACTIF, EN ATTENTE, PRET. L'état actif correspond alors à la mise en oeuvre par le module d'un traitement adapté à l'évènement : par exemple, remises à zéro des commandes, ... Le passage à l'état PRET est assuré par les signaux PAUSE et RESET issus du moniteur,

- si la tâche peut être abandonnée sans précaution, le module moniteur génère directement les signaux PAUSE et RESET.

Certains dialogues intertâches ne nécessitent pas forcément de telles transitions ; pour y satisfaire, il est alors possible d'utiliser les interruptions esclave → moniteur et moniteur → esclave, en ayant pris soin de préciser au préalable, par l'intermédiaire d'indicateurs adéquats placés en mémoire commune, qu'il s'agit d'une procédure particulière et non d'une transition.

A notre avis, les différents états envisagés doivent permettre de répondre à la majorité des problèmes rencontrés en conduite de processus en temps réel.

Compte tenu des principes précédemment définis, nous avons réalisé un calculateur multimicroprocesseurs autorisant la conduite en parallèle de 8 tâches maximum, ce qui nous a paru suffisant pour les applications visées.

C H A P I T R E I I

REALISATION DU CALCULATEUR MULTIMICROPROCESSEURS

II-1) INTRODUCTION

La réalisation du calculateur multimicroprocesseurs a été guidée par un certain nombre de préoccupations :

- obtenir un calculateur avec un minimum de cartes différentes afin de réduire les frais d'étude des circuits imprimés et de faciliter l'interchangeabilité des modules,

- utiliser un minimum de matériel hautement intégré pour une réalisation aisée du circuit imprimé, le maximum de fonctions étant confié au logiciel. Ceci permet de réaliser des cartes à coût modique et de bonne fiabilité,

- pouvoir connecter un système de développement qui permette au concepteur la mise au point aisée de son logiciel et assure à l'utilisateur une maintenance rapide par la localisation automatique de l'élément défaillant,

- pouvoir s'intégrer dans une commande hiérarchisée autorisant une extension verticale du système.

II-2) CHOIX DU MICROPROCESSEUR

Les critères qui ont guidés notre choix sont :

- huit bits sont suffisants, car la plupart des données à traiter dans une commande de processus sont logiques, ou d'amplitude limitée nécessitant rarement des calculs complexes,

- étant donné la répartition en modules, la capacité d'adressage n'est pas une contrainte,

- l'utilisation du microprocesseur que nous envisageons, nécessite de disposer de signaux simples à interpréter afin de limiter l'environnement nécessaire,

- obtenir, si possible, un calculateur utilisant une source de tension unique (+ 5 V) afin d'en réduire le prix.

Ces considérations nous ont amené à choisir le microprocesseur 2650 de la Société SIGNETICS. Ce microprocesseur ne nécessite qu'une horloge monophasée et autorise 64 interruptions vectorisées.

A noter que ce choix a été fait, il y a trois ans et que depuis des microprocesseurs plus sophistiqués sont arrivés sur le marché. Mais nous verrons, par la suite, que ce choix n'est pas unique et que d'autres microprocesseurs peuvent s'intégrer dans le calculateur.

II-3) PRESENTATION GENERALE DU CALCULATEUR

Le calculateur se compose de 2 types de cartes :

- la carte microcalculateur qui est utilisée indifféremment comme module MONITEUR ou module ESCLAVE. Seuls trois circuits intégrés montés sur supports permettent de passer d'un module à un autre. Chaque module constitue un microcalculateur capable de fonctionner de façon autonome,

- la carte mémoire commune "boîte à lettre" ; qui est équipée de mémoire RAM sauvegardée ou non.

Nous n'avons pas développé de cartes d'entrée-sortie universelles car elles auraient été sous-utilisées. Nous avons préféré générer des signaux rendant la connection des cartes d'entrée-sortie simple et rapide.

La figure II.1. donne un synoptique du calculateur réalisé en version semi-industrielle, où nous n'avons représenté qu'un module ESCLAVE, mais le système est capable d'en accueillir huit.

Les modules microcalculateur, tant MONITEUR qu'ESCLAVE, développent deux bus distincts :

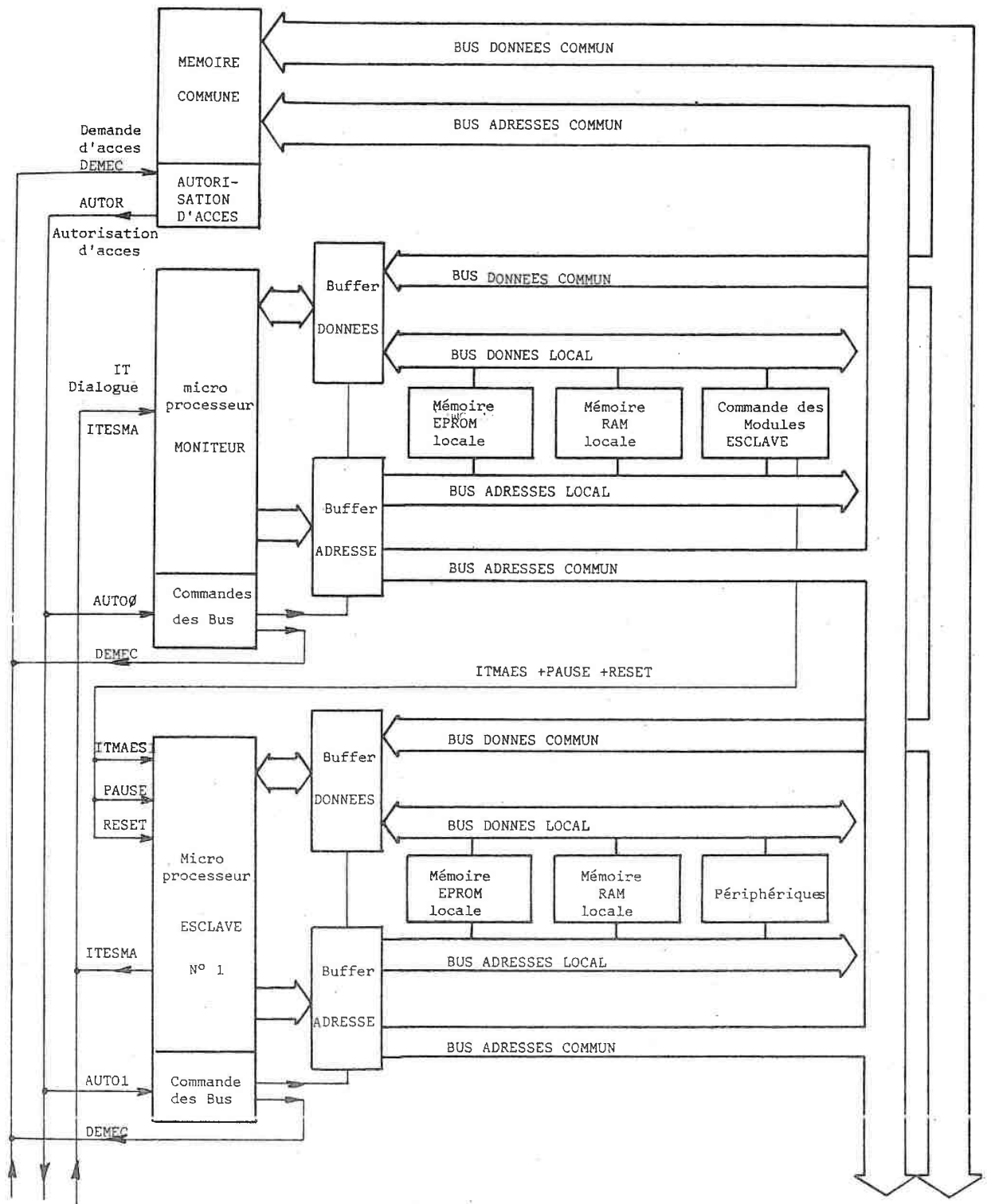


Figure II.1- Organisation générale du calculateur

- le bus LOCAL qui adresse la mémoire locale EPROM (Erasable Programmable Read Only Memory) et RAM (Random Acces Memory) et les registres de commandes affectés aux périphériques,
- le bus COMMUN qui permet d'atteindre la mémoire commune et auquel tous les microcalculateurs sont connectés.

La logique d'accès à la mémoire "boîte à lettre" est répartie sur la carte mémoire commune et sur les cartes micro. La mémoire commune donne l'autorisation d'accès séquentiellement à chaque microcalculateur qui peut alors effectuer un accès.

Les 3 signaux de dialogue module MONITEUR vers module ESCLAVE sont :

- l'interruption (ITMAES) permettant de faire sortir un microprocesseur de l'état "EN ATTENTE",
- la mise en pause (PAUSE) qui arrête le microprocesseur lorsqu'il a terminé l'instruction en cours de traitement,
- le lancement à l'adresse zéro (RESET) qui est utilisé pour la transition PRET → ACTIF.

Le dialogue modules ESCLAVE vers module MONITEUR est assuré par les signaux (ITESMA) qui sont regroupés en "OU FANTOME" pour créer une interruption au module MONITEUR.

Nous présentons en détail dans les paragraphes suivants les différents éléments constituant le calculateur.

Les principales caractéristiques et les signaux du microprocesseur 2650 sont donnés en annexe technique.

II-4) PRESENTATION DES DIFFERENTS ELEMENTS DU MICROCALCULATEUR

La figure II.2. situe les éléments implantés sur une carte microcalculateur.

II-4-1) Bus locaux et communs

Le bus de données et le bus d'adresse générés par le microprocesseur ont été dédoublés afin de former deux bus distincts affectés l'un au fonctionnement local et l'autre à l'accès à la mémoire commune.

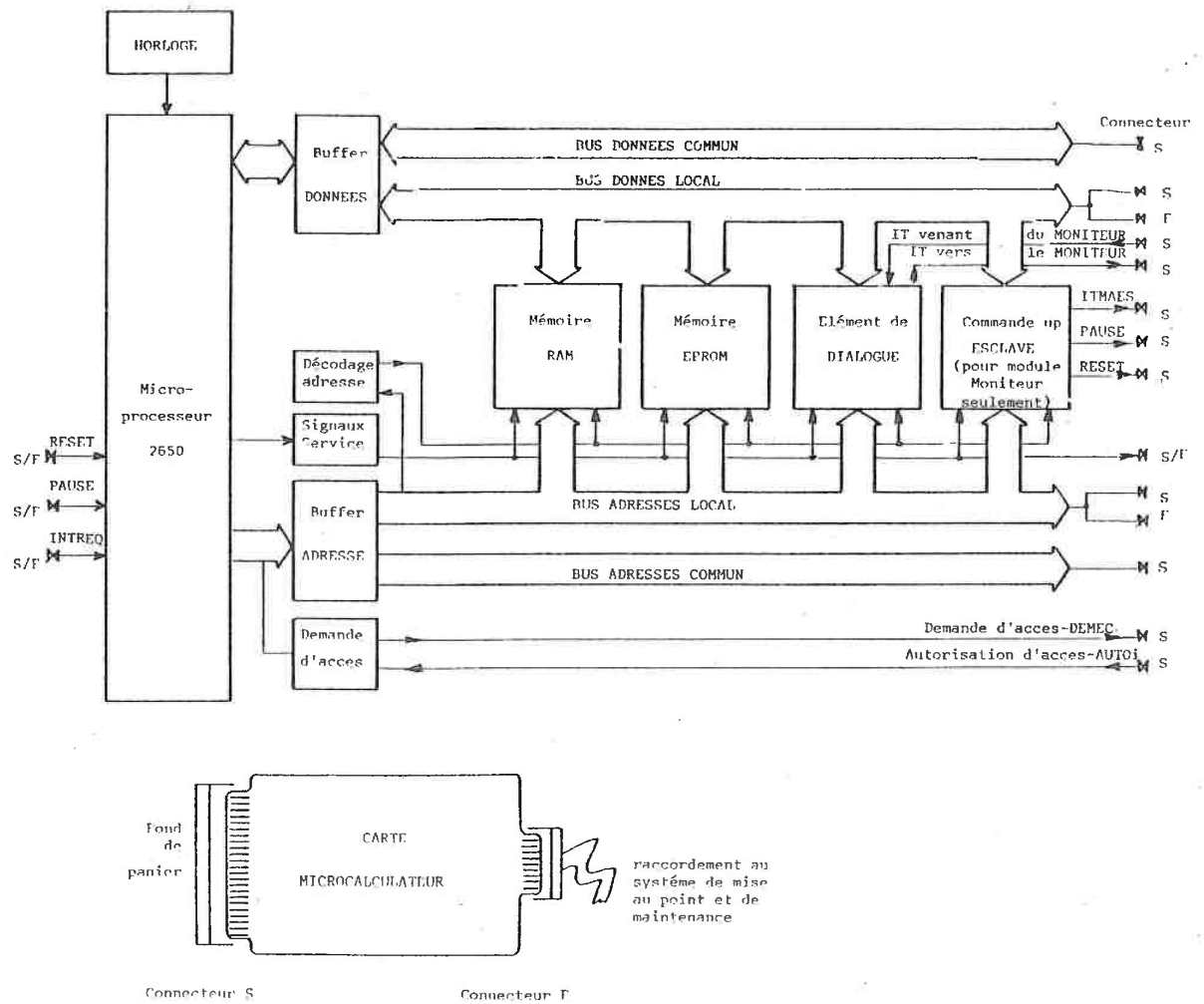


Figure II.2- Organisation générale d'une carte microcalculateur.

Ces bus sont en haute impédance tant que le microprocesseur ne fait pas une demande d'opération externe et sont activés par le signal OPREQ (opération request) qui coordonne les opérations destinées à la mémoire ou aux registres d'Entrée-Sortie.

Lorsque le microprocesseur est en HALT, les bus d'adresse et de données sont donc en haute impédance et il est alors possible d'avoir accès à la mémoire locale et aux registres des périphériques par un élément extérieur, tel qu'un pupitre de mise au point ou un calculateur situé en amont.

II-4-2) Signaux de service locaux

Après analyse des besoins des éléments extérieurs au microprocesseur, on dénombre quatre signaux de service permettant de gérer la mémoire locale et les différents périphériques (figure II.3.).

Ces quatre signaux précisent :

- l'affectation du transfert,

$M/\overline{IOL}^* = 0$ avec la mémoire,

$\overline{M}/IOL^* = 0$ avec un périphérique,

- la nature du transfert,

$R/\overline{WL}^* = 0$ pour une lecture,

$= 1$ pour une écriture,

- et le moment d'échantillonnage lors d'une écriture. Ce signal WRP, issu du microprocesseur, est une impulsion de 400 ns centrée sur OPREQ que nous utilisons pour écrire dans les différents registres de sortie.

Pour que la mémoire locale et les divers périphériques attachés au microcalculateur puissent être accessibles par un pupitre de chargement et de mise au point, il faut que celui-ci puisse également gérer ces signaux de service.

Afin de permettre cette connection, nous commandons les signaux issus du microcalculateur par des portes à collecteur ouvert validées par le signal ACKLOC lorsque la mémoire locale ou un registre de périphérique est adressé.

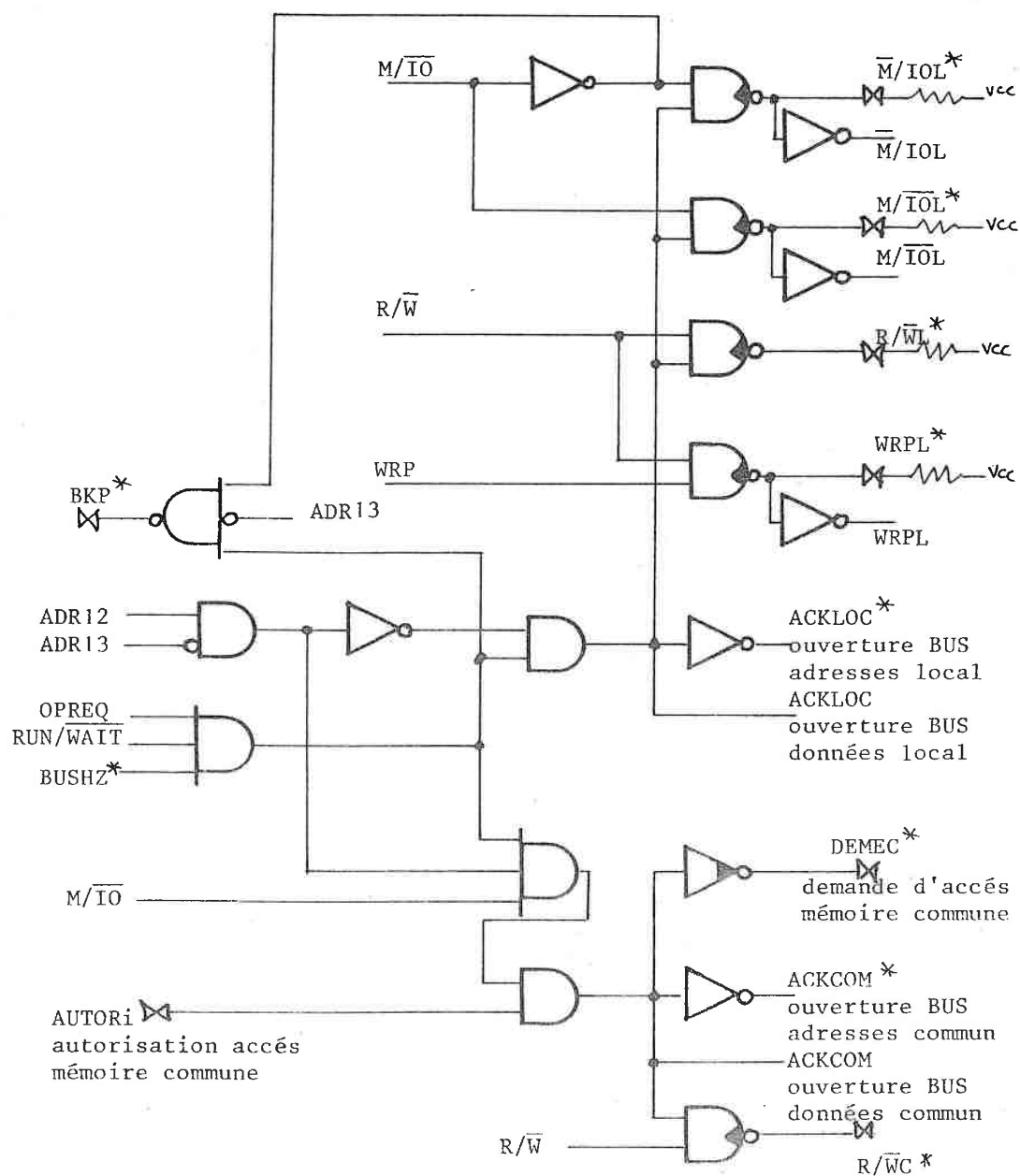


Figure II.3- Signaux de service

II-4-3) Signal de service commun

Le seul signal de service nécessaire pour effectuer un accès à la mémoire commune est $R/\overline{WC}^*$ qui précise si l'on effectue une lecture ou une écriture. Tous les signaux $R/\overline{WC}^*$ issus des divers microcalculateurs sont regroupés à travers des portes à collecteur ouvert.

Lorsqu'une demande d'accès à la mémoire commune est faite et qu'elle est autorisée (cf. paragraphe II-6), le signal ACKCOM devient actif et ouvre la porte de commande du signal $R/\overline{W}$.

II-4-4) Horloge

L'horloge nécessaire au fonctionnement séquentiel du microprocesseur est simplement réalisée par un circuit 74123 contenant 2 monostables réarmables dont l'un fixe la fréquence (1 MHz) et l'autre la largeur de l'impulsion (500 ns).

II-4-5) Décodage des adresses mémoire

La capacité d'adressage mémoire du microprocesseur 2650 est de 32 Koctets. Nous avons limité l'espace adressable depuis le pupitre de mise au point à 16 Koctets, car l'utilisation du bit d'adresse de poids le plus fort comme discriminateur permet de considérer la mémoire du calculateur comme le prolongement de la mémoire du pupitre de mise au point et allège la mise en oeuvre.

Nous pensons que cette capacité de 16 Koctets par microcalculateur est suffisante dans une structure multiprocesseurs. On peut noter que les 16 Koctets, non accessibles par le pupitre, sont néanmoins utilisables par le microcalculateur.

Nous avons découpé l'espace mémoire en 3 blocs :

- 1 bloc de 4 Koctets réservé à la mémoire locale et implanté sur chaque carte microcalculateur. Il se compose de :
 - . 2 Koctets de EPROM (adresse 0 à 7FF),
 - . 1 Koctet de EPROM (adresse 800 à BFF),
 - . 1 Koctet de RAM (adresse C00 à FFF),

- 1 bloc de 4 K octets affectés à la mémoire commune (adresse 1000 à 1FFF),
- 1 bloc de 8 K octets permettant une extension de la mémoire locale sur une carte annexe. (adresse de 2000 à 3FFF).

L'utilisation judicieuse du décodeur BCD-décimal (7442) permet une sélection simple et rapide des différents circuits mémoire comme le montre la figure II.4.

Ainsi, pour la mémoire implantée sur la carte microcalculateur, le décodage des bits d'adresse de poids 10, 11 et 12 n'est autorisé que lorsque l'on adresse la mémoire locale et que le bit d'adresse de poids 13 est à zéro. Au contraire, ce bit devra être à 1 pour permettre le décodage des 8 K octets de la carte d'extension.

Le décodage des 4 K octets affectés à la mémoire commune est autorisé lorsqu'une demande d'accès à la mémoire commune est faite.

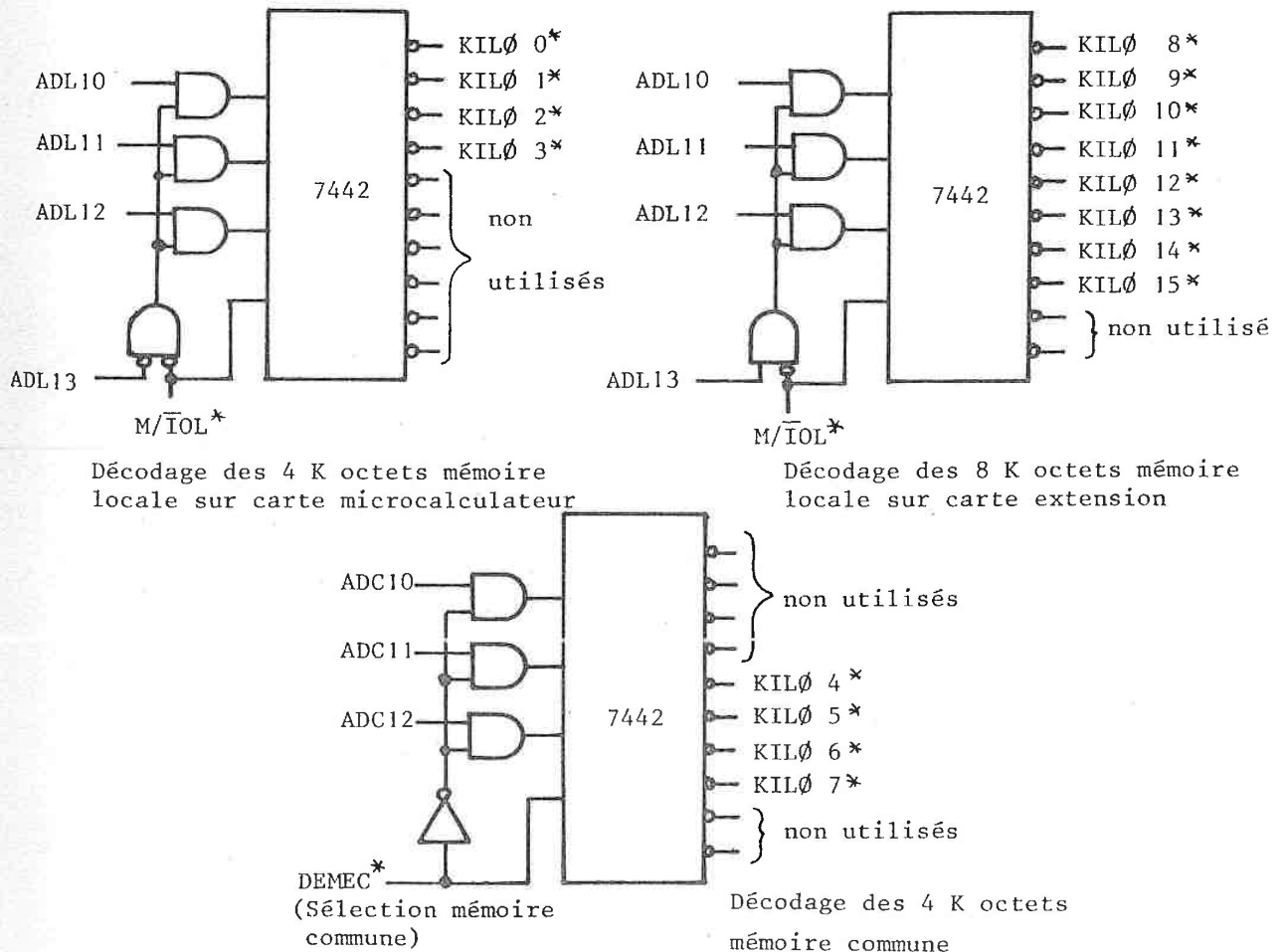


Figure II.4. : Décodage des blocs mémoires.

II-4-6) Décodage des adresses des registres d'entrée-sortie

L'adresse du périphérique à atteindre, lors de l'exécution d'une instruction d'entrée-sortie en mode étendu, est codée sur les 8 premiers bits d'adresse. Ceci permet d'adresser 256 registres d'entrée-sortie mais nécessite un décodage d'adresse assez lourd. Afin de limiter le matériel, nous avons affecté le bit d'adresse 2° au registre "DIALOGUE" qui autorise ou génère les interruptions de dialogue entre Moniteur et Esclave (cf. paragraphe II-4-8), et réservé les 7 bits restants aux registres d'entrées-sorties.

Si, pour un microcalculateur ESCLAVE, le nombre de registres nécessaires est inférieur à 18, on utilise le décodage suivant qui met en oeuvre un décodeur BCD-décimal 7442 (figure II.5.). Etant donné que le décodage est incomplet, seules les adresses ci-dessous (tableau II.5.bis) peuvent être utilisées sous peine d'adressages multiples.

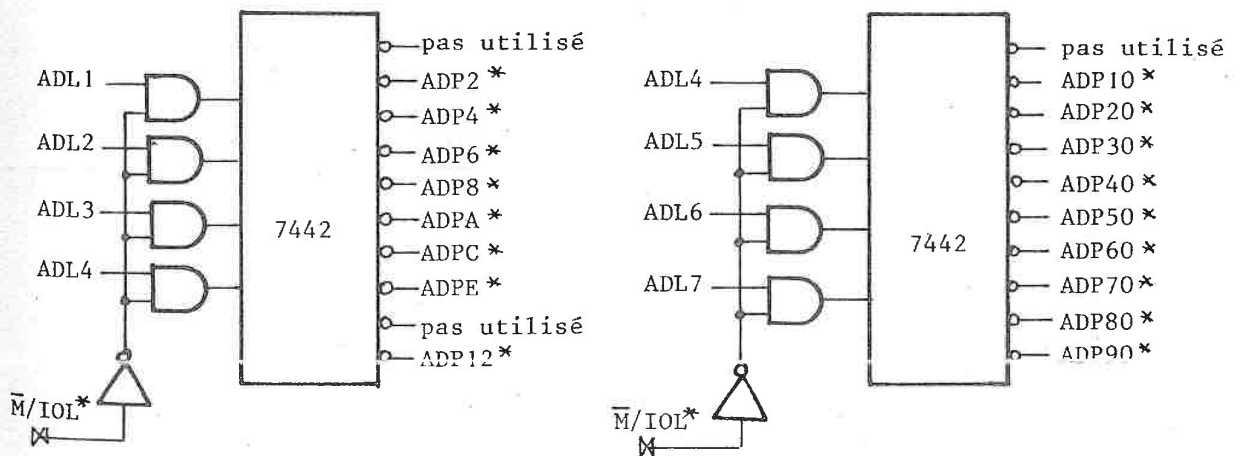


Figure II.5. Décodage des adresses des périphériques

Adresse binaire	Adresse	Adresse binaire	Adresse
7 6 5 4 3 2 1 0	mnémonique	7 6 5 4 3 2 1 0	mnémonique
0 0 0 0 0 0 0 1	ADP1 Dialogue	0 0 0 1 0 0 0 0	ADP10
0 0 0 0 0 0 1 0	ADP2	0 0 1 0 0 0 0 0	ADP20
0 0 0 0 0 1 0 0	ADP4	0 0 1 1 0 0 0 0	ADP30
0 0 0 0 0 1 1 0	ADP6	0 1 0 0 0 0 0 0	ADP40
0 0 0 0 1 0 0 0	ADP8	0 1 0 1 0 0 0 0	ADP50
0 0 0 0 1 0 1 0	ADPA	0 1 1 0 0 0 0 0	ADP60
0 0 0 0 1 1 0 0	ADPC	0 1 1 1 0 0 0 0	ADP70
0 0 0 0 1 1 1 0	ADPE	1 0 0 0 0 0 0 0	ADP80
0 0 0 1 0 0 1 0	ADP12	1 0 0 1 0 0 0 0	ADP90

Tableau II.5.bis. : Adresses des périphériques.

Si l'on désire augmenter le nombre d'Entrées-Sorties, il est alors nécessaire de combiner ces adresses.

II-4-7) Mémoire locale

Les mémoires que nous avons choisies sont un compromis entre le degré d'intégration et le prix, avec la condition de requérir qu'une tension de 5 volts.

Nous utilisons de la mémoire EPROM 2716 de 2 K octets monotension et de la mémoire RAM 2114L de 4 Kilobits.

II-4-8) Eléments de dialogue

La procédure de dialogue entre microcalculateur MONITEUR et microcalculateur ESCLAVE est essentiellement basée sur l'échange d'interruptions. Lorsqu'un module ESCLAVE veut dialoguer avec le module MONITEUR, il initialise un indicateur en mémoire commune à une place fixe et génère une interruption à destination du MONITEUR. Toutes les interruptions issues des microcalculateurs ESCLAVE sont regroupées par l'intermédiaire de portes logiques à collecteur ouvert. Dans le sous-programme d'interruption, le module MONITEUR teste les différents indicateurs par ordre de priorité décroissante et traite

la demande correspondante, après avoir remis à zéro l'indicateur. La sortie du sous-programme d'interruption ne se fait que lorsque toutes les demandes ont été satisfaites, afin de ne pas perdre une interruption que serait arrivée pendant le traitement du sous-programme.

Lorsque le module MONITEUR veut dialoguer avec un module ESCLAVE, il envoie une interruption à destination de celui-ci.

Le système de prise en compte des interruptions de dialogue se compose d'une bascule de mémorisation avec sa remise à zéro, logicielle et matérielle, (figure II.6.), d'une bascule de masquage et d'un circuit à sortie 3-états permettant de déposer le vecteur d'interruption sur le bus de données locales à l'arrivée du signal INTACK*, (Interrupt acknowledge).

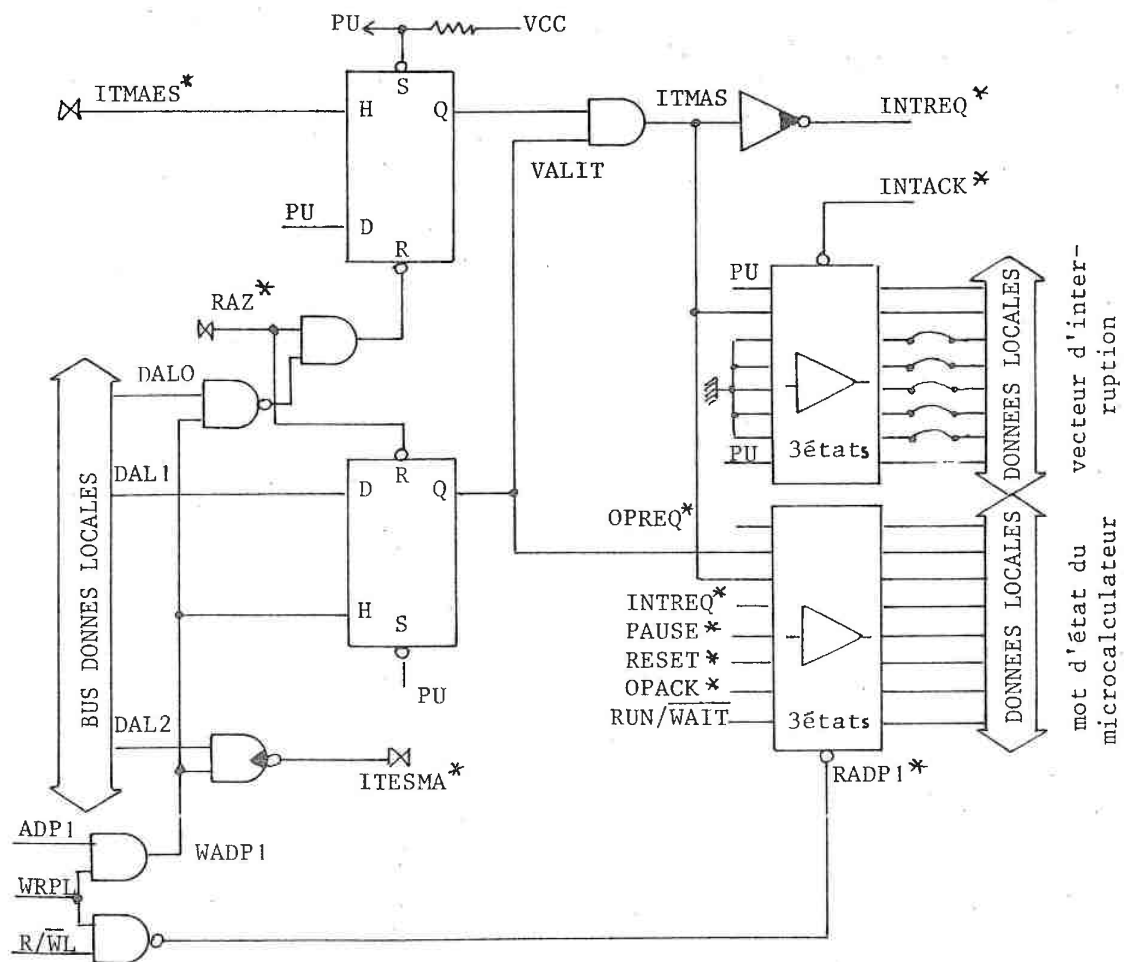


Figure II.6.

Les "straps", en sortie du circuit de génération de vecteur, permettent de câbler des vecteurs d'interruption différents sur les diverses cartes d'Entrée-Sortie.

II-4-9) Système d'interruption

Dans la conduite d'un processus industriel, le système d'interruption doit être bien adapté, car c'est lui qui permet la commande en "temps réel".

Voyons d'abord les possibilités du 2650 sur ce point.

II-4-9-1) Procédure d'interruption du 2650

Le microprocesseur 2650 possède un seul niveau d'interruption vectorisé, chaque périphérique interrompant doit donc fournir l'adresse de la séquence à effectuer. D'autre part, il est possible, par programme, d'inhiber ou d'autoriser le traitement de l'interruption grâce au bit II (inhibit interrupt) du mot d'état du processeur.

Les adresses de retour sont sauvegardées dans une pile matérielle qui n'autorise que 8 imbrications de sous-programmes au maximum.

Le déroulement d'une séquence d'interruption est la suivante :

- le programme autorise l'interruption, (bit II = 0),
- la demande d'interruption est générée par un périphérique en agissant sur la ligne INTREQ* (Interrupt request),
- le processeur termine l'instruction en cours, sauvegarde la valeur du compteur ordinal dans la pile,
- le processeur interdit automatiquement les interruptions futures en mettant à 1 le bit II du mot d'état,
- le processeur force le code opération de l'instruction ZBSR (branchement relatif à un sous-programme par rapport à l'adresse zéro) dans le registre instruction,
- le processeur fournit l'accusé de réception à la demande par le signal INTACK* (Interrupt acknowledge) et échantillonne l'adresse de branchement sur le bus de données, fournie par le périphérique interrompant.

Le bit de poids fort de cette adresse peut indiquer une indirection ; dans ce cas les 7 bits restants permettent de vectoriser 64 interruptions.

II-4-9-2) Génération de l'adresse du vecteur

Nous avons jugé que 64 interruptions n'étaient pas nécessaires dans une structure multiprocesseurs et nous avons limité leur nombre à six, une interruption étant réservée au dialogue moniteur-esclave et les cinq restantes affectées aux périphériques. Cette restriction permet de résoudre, avec un minimum de matériel, le problème de la hiérarchisation en présence de plusieurs interruptions simultanées.

En effet, nous avons choisi d'affecter à chaque demande d'interruption un bit de l'adresse du vecteur qui est donc constituée de la manière suivante :

- bit 0 est forcé à 1 afin que la première adresse de vecteur soit 3, car les trois premières places mémoire sont réservées au lancement du microprocesseur,
- bit 1 est réservé à l'interruption de dialogue avec le moniteur,
- les bits 2 à 6 sont affectés chacun à un périphérique,
- le bit 7 est forcé à 1 pour que cette adresse soit interprétée comme une adresse indirecte.

L'interruption dialogue étant inhérente au microcalculateur, nous avons implanté sur la carte microcalculateur le circuit de génération de l'adresse de ce vecteur muni de 5 "straps" (figure II.7.).

Pour brancher une interruption externe, il suffit d'ôter le "strap" relatif au bit qui lui est affecté ; ce bit étant fourni par la carte d'entrée-sortie correspondante.

Les adresses des vecteurs d'interruption sont donc :

- adresse 3 pour le dialogue Moniteur ↔ Esclaves,
 - adresse 5
 - adresse 9
 - adresse 11
 - adresse 21
 - adresse 41
- } pour les périphériques.

Lorsque plusieurs interruptions se présentent simultanément, l'adresse du vecteur est alors composée des différents bits correspondant aux demandes.

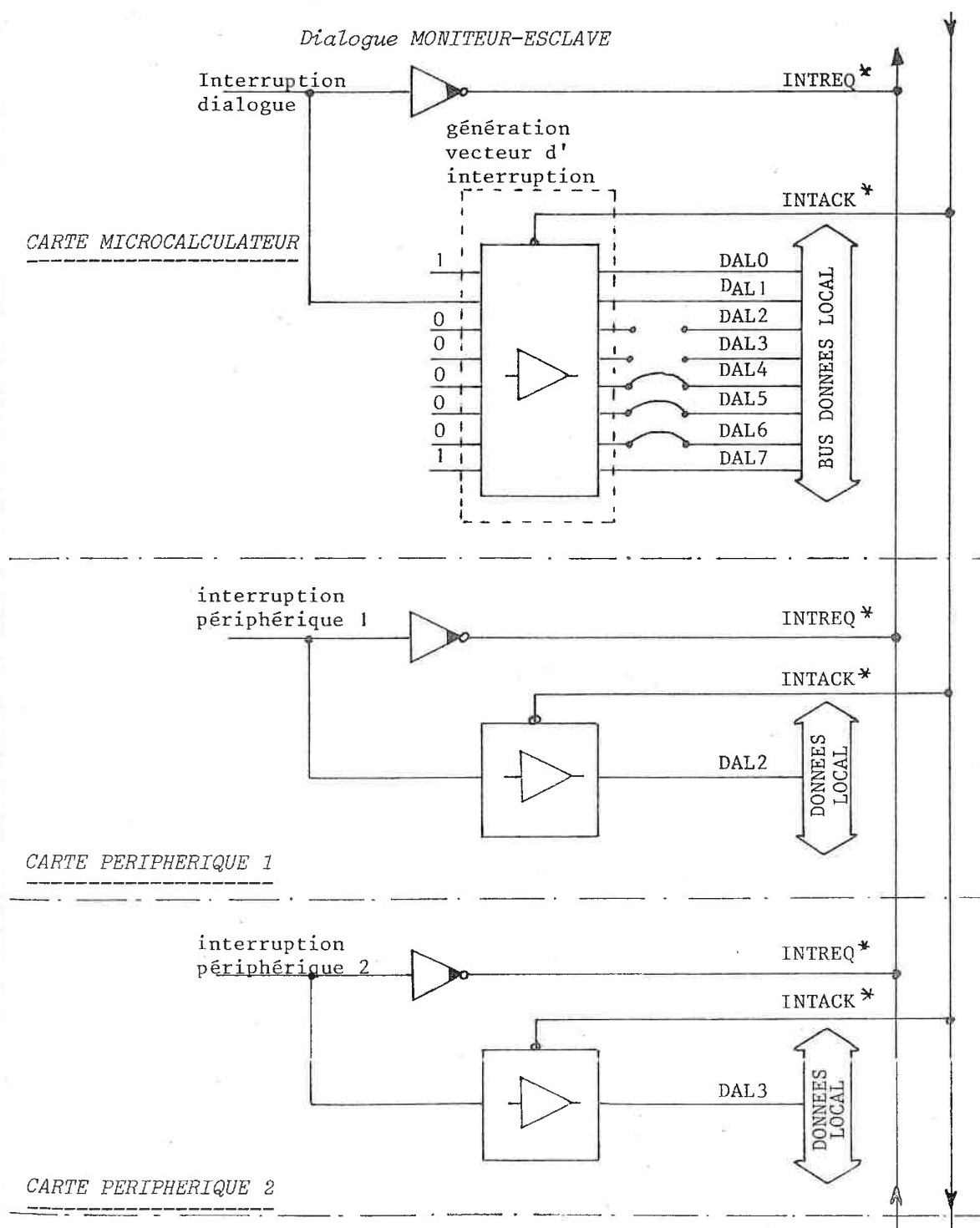


Figure II.7.- Principe de génération de vecteur d'interruption.
Exemple pour deux interruptions périphériques.

Le vecteur, dont l'adresse est ainsi générée, est donc relatif à l'ensemble des demandes simultanées. L'adresse de déroulement, que ce vecteur doit contenir, est celle du sous-programme d'interruption le plus prioritaire, ce qui permet une hiérarchisation des traitements.

Le tableau II.8. montre les 7 possibilités de vecteurs pour 3 interruptions différentes et les conflits qui pourraient apparaître.

Adresse du vecteur	Contenu du vecteur
03	Adresse sous-programme Maître-esclave
05	Adresse sous-programme périphérique 1
07	Adresse sous-programme dialogue ou Adresse sous-programme périphérique 1 suivant priorité
09	Adresse sous-programme périphérique 2
0B	Adresse sous-programme dialogue ou Adresse sous-programme périphérique 2 suivant priorité
0D	Adresse sous-programme périphérique 1 ou Adresse sous-programme périphérique 2 suivant priorité
0F	Adresse sous-programme dialogue ou Adresse sous-programme périphérique 1 ou Adresse sous-programme périphérique 2 suivant priorité

Tableau II.8. : Table des vecteurs d'interruption pour deux périphériques.

L'utilisation de 126 places mémoire permet donc de hiérarchiser simplement le traitement de 6 interruptions, et autorise une hiérarchisation dynamique si la mémoire est vive.

II-4-9-3) Temps de sauvegarde du contexte

Lors de l'arrivée d'une interruption, le processeur ne sauvegarde que la valeur du compteur ordinal, il est donc nécessaire de préserver la valeur des variables du processeur (registres et mot d'état) afin de pouvoir restituer le contexte du programme interrompu après traitement de l'interruption.

Si la sauvegarde de tous les registres n'est pas toujours nécessaire, le mot d'état du processeur (PSL) contenant le contexte arithmétique et logique doit être impérativement sauvegardé. Pour cela, l'entrée dans un sous-programme d'interruption se fait en exécutant 3 instructions destinées à stocker la valeur du registre banalisé RO et du PSL en mémoire vive (figure II.9.). La sortie du sous-programme d'interruption passe par l'exécution de 4 instructions de restitution du contexte. La prise en compte d'une interruption est au maximum de 9,6 μ s et la sauvegarde du contexte demande 24 μ s.

La restitution du contexte doit se trouver obligatoirement en mémoire vive puisque l'on effectue une modification d'instruction au moment de la sauvegarde.

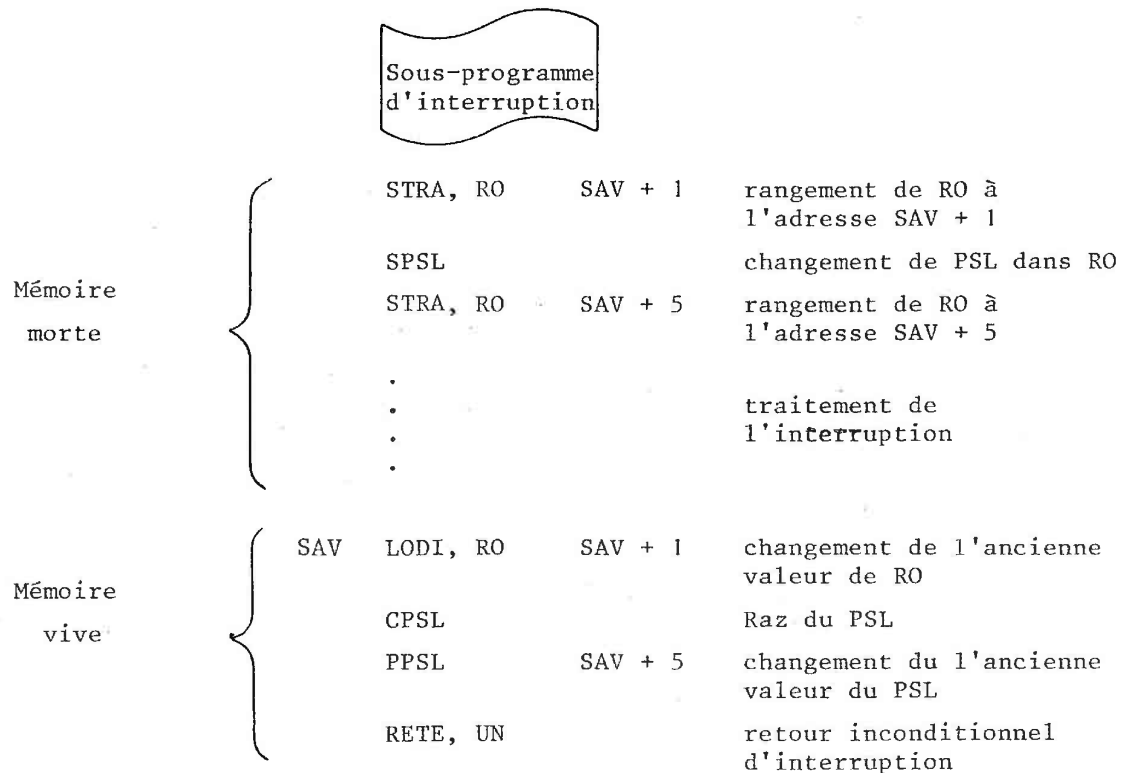


Figure II.9.

II-4-10) Mise en attente du processeur

Les périphériques du processeur, mémoire ou entrées-sorties, peuvent avoir des temps d'accès différents ou se déclarer occupés. C'est le cas en particulier lors d'une demande d'accès à la mémoire commune qui peut être en transfert avec un autre microcalculateur. La mise en attente du processeur est possible par le signal OPACK* (opération acknowledge). En laissant ce signal à 1, on bloque le processeur dans sa phase de transfert externe et celui-ci maintient les données, les adresses et les signaux de service actifs. Le signal OPACK* est scruté par le front descendant de l'horloge et lorsque le processeur trouve OPACK* à 0, il continue son cycle. Pour gérer OPACK* nous utilisons deux signaux "OCCUPE", l'un pour les transferts locaux (OCCL) et l'autre généré par la mémoire commune (OCCC).

II-4-11) Mot d'état du microcalculateur

Pour faciliter le diagnostic logiciel d'une défaillance du microcalculateur, il faut disposer d'un mot d'état du microcalculateur. L'expérience nous a montré que huit signaux suffisent à définir l'état de celui-ci. Ces signaux sont (se reporter à la figure II.6.) :

- RUN/WAIT : marche ou attente,
- INTREQ* : demande d'interruption,
- PAUSE* : mise en pause,
- RESET* : initialisation du compteur ordinal,
- OPREQ* : demande opération externe,
- OPACK* : réponse à une demande externe,
- VALIT : validation de l'interruption dialogue,
- ITMAS : interruption de dialogue.

II-5) INSERTION DU CALCULATEUR DANS UNE STRUCTURE HIERARCHISEE

Le dialogue avec un ordinateur situé en amont dans une structure hiérarchisée peut être effectué suivant deux modes :

- couplage asynchrone parallèle puisque l'on dispose de tous les signaux utiles à l'accès direct mémoire,
- couplage asynchrone série en adjoignant un coupleur spécifique.

II-6) MEMOIRE COMMUNE

La mémoire commune, seule ressource commune, stocke les indications utiles au dialogue esclave-moniteur et sert de "boîte à lettre" entre les différents microcalculateurs. L'utilisation d'une telle mémoire nécessite un allocateur pour lever les conflits dans le cas de demandes d'accès simultanées par plusieurs modules. Cet allocateur peut être réparti entre les divers calculateurs ou centralisé au niveau de la mémoire commune.

II-6-1) Allocateur

Dans un allocateur réparti, toutes les demandes d'accès sont regroupées et un signal de retour transitant par chaque microcalculateur recherche celui qui a demandé.

Nous avons représenté schématiquement le principe (figure II.10.).

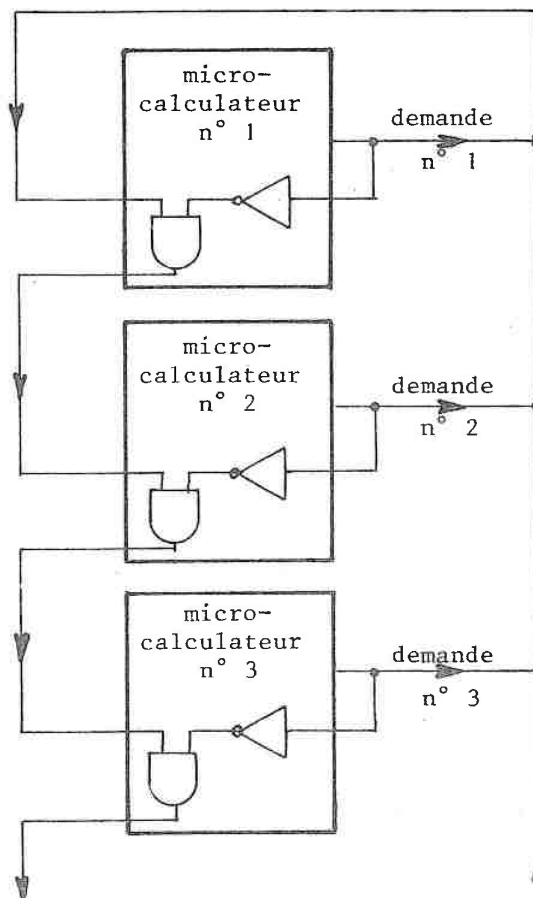


Figure II.10. : Principe d'un allocateur réparti.

Lorsqu'un microcalculateur ne fait pas de demande, il laisse transiter le signal qui est envoyé sur le module suivant. Le module demandeur bloque le signal et accède à la mémoire commune.

Dans cette architecture, c'est l'emplacement géographique du microcalculateur qui fixe sa priorité.

Bien que simple dans le principe, la réalisation d'un allocateur réparti demande plus de matériel qu'un allocateur centralisé, c'est pourquoi, nous avons retenu cette deuxième solution.

L'allocateur centralisé, qui se situe sur la carte mémoire commune, teste chaque microcalculateur pour savoir s'il a un transfert à effectuer. Dans l'affirmative, il arrête sa scrutation et accorde le bus commun au module demandeur. Après le transfert, la suite de la séquence de scrutation peut être choisie par commutateur :

- soit, l'allocateur continue la scrutation là où il était arrêté,
- soit, l'allocateur reprend la scrutation au début.

Le choix entre ces deux modes de fonctionnement dépend des transferts à effectuer avec la mémoire commune.

La première solution n'accorde de priorité à aucun microcalculateur, la seconde définit une priorité en fonction de l'emplacement géographique de la carte microcalculateur.

La réalisation de cet allocateur centralisé demande une horloge et un compteur qui charge un registre à glissement. Les sorties de ce registre, représentées figure II.11., sont envoyées aux différents microcalculateurs.

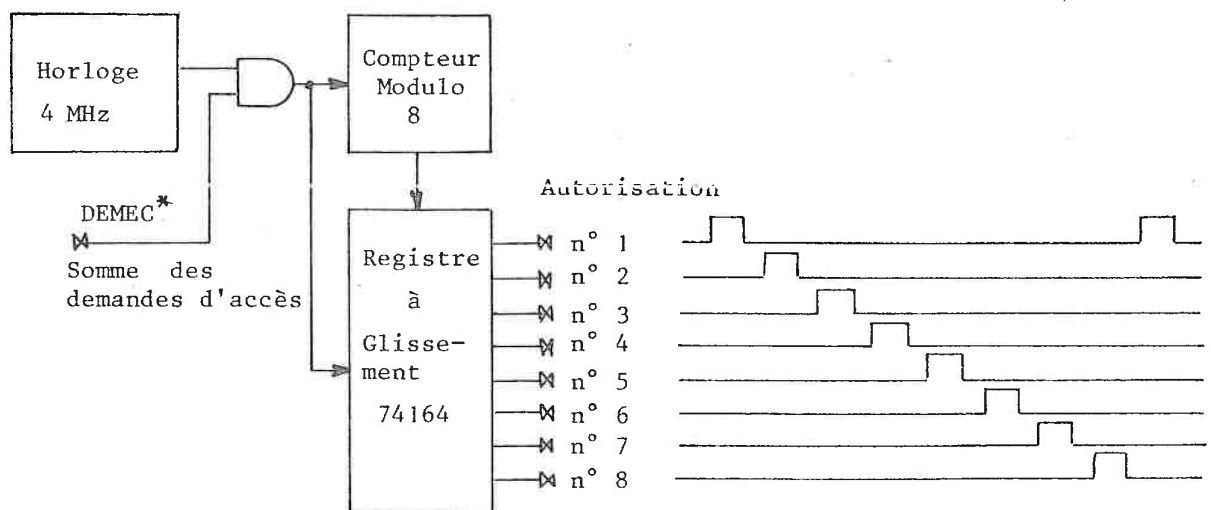


Figure II.11.

Avec une horloge à 4 MHz, il faut en moyenne 2 μ s pour qu'un microcalculateur accède à la mémoire commune, ce qui représente une attente de 15 % par rapport au temps d'exécution de l'instruction de transfert.

II-6-2) Type de mémoire utilisée

Si la mémoire commune doit conserver ses informations en cas de coupure ou de micro-coupure de courant, il est nécessaire d'utiliser des mémoires du type CMOS basse consommation associées à une batterie pour assurer la rétention des informations.

Le type de mémoire qui nous semble le mieux adapté actuellement est la mémoire 5101L qui nécessite une tension de 2 v en mode "standby" avec une consommation de 10 microampères, ce qui permet avec deux piles R20 de 1,5 v de maintenir les informations pendant plusieurs semaines ou plusieurs mois suivant la capacité de la mémoire commune.

II-6-3) Temps d'attente des données

Lorsqu'une place mémoire est sélectionnée, il faut un certain temps avant de disposer de son contenu : c'est le temps d'accès, qui dépend de chaque type de mémoire.

D'autre part, certaines mémoires ont une chronologie à respecter lors de l'écriture d'une donnée. La chronologie des séquences fait apparaître trois temps, après la sélection du bloc mémoire concerné :

- temps de sélection de la place mémoire,
- temps d'écriture dans cette place,
- temps de maintien qui est généralement court.

La mémoire CMOS que nous utilisons demande un temps de sélection de 100 ns minimum, un temps d'écriture de 450 ns minimum et un temps de maintien de 70 ns.

Nous avons donc créé cette chronologie avec des monostables dont le premier est enclenché par l'apparition du signal de demande d'accès DEMEC*. Après ce temps, la mémoire se déclare libre et relâche le signal mémoire commune occupée (OCCC) (cf. II-4-10).

II-6-4) Interchangeabilité des microprocesseurs

La seule contrainte à l'utilisation d'un type différent de microprocesseur est la possibilité de mise en attente illimitée avec maintien des adresses et des données lorsque l'élément extérieur se déclare occupé.

Ceci est indispensable pour les transferts en mémoire commune puisque celle-ci est la seule ressource commune à tous les microcalculateurs.

Les autres signaux utilisés (RESET, PAUSE) sont toujours disponibles sur les microprocesseurs actuellement commercialisés.

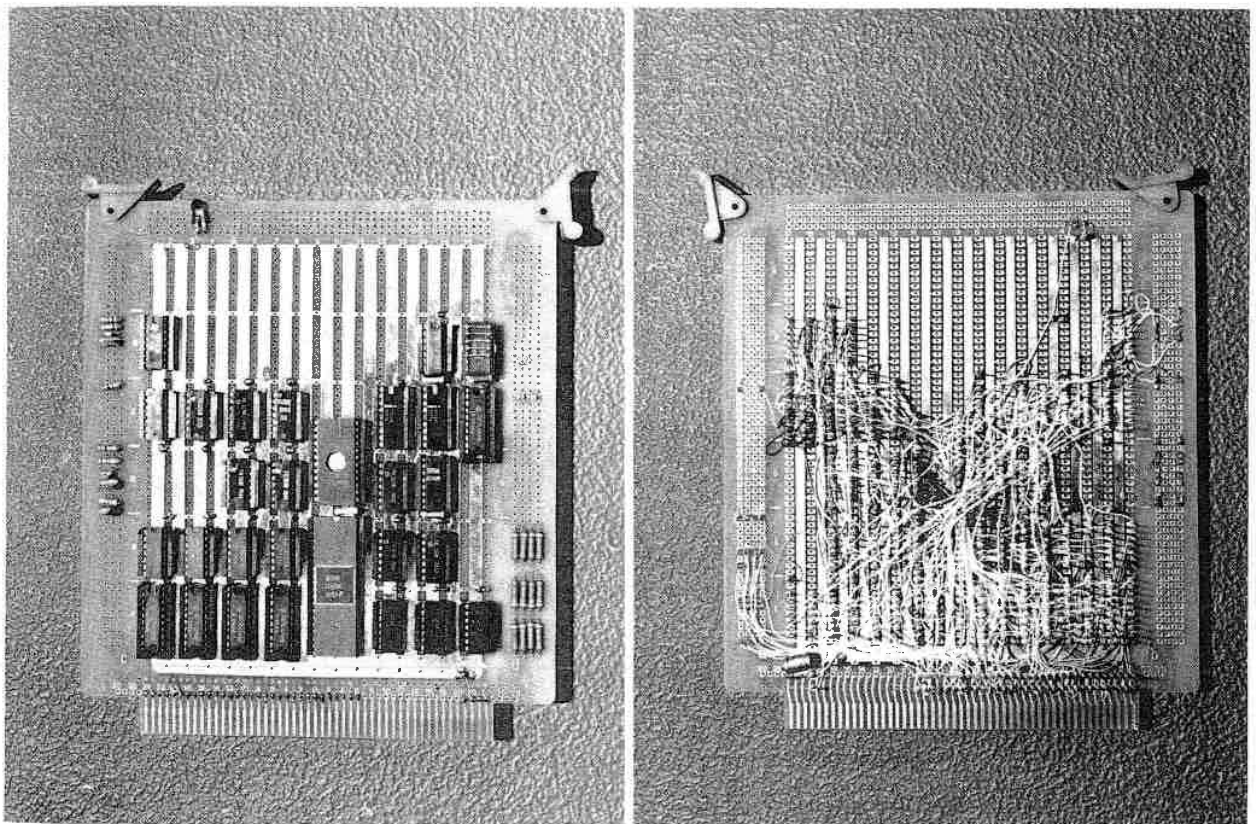


Figure II.12- Carte microcalculateur, version semi-industrielle, réalisée en miniwrapping.

CHAPITRE III

SYSTEME DE MISE AU POINT ET DE MAINTENANCE

III-1) INTRODUCTION

Etant donné que ce calculateur sera réalisé et utilisé par des industriels non électroniciens qui devront en assurer la maintenance, nous avons donc développé un système pour tester le bon fonctionnement des cartes microcalculateurs après fabrication, permettre une mise en oeuvre rapide et facile et aider à la localisation d'une liaison ou d'un module défectueux.

Après étude, il s'avère que ces trois possibilités peuvent cohabiter et utiliser le même matériel à la condition de pouvoir accéder simultanément à tous les microcalculateurs.

En effet, il est indispensable dans la phase finale de mise au point, de pouvoir contrôler tous les microcalculateurs simultanément et de pouvoir accéder à la mémoire commune. D'autre part, l'éventail des tests de maintenance est considérablement augmenté si on a la possibilité de contrôler l'émission et la réception des signaux entre microcalculateurs.

Des programmes spécifiques permettront alors d'utiliser le système comme outil de développement, comme testeur de cartes ou comme assistant dans la recherche d'une panne.

Nous présentons dans le paragraphe suivant la version prototype de ce système de mise au point et de maintenance.

III-2) VERSION PROTOTYPE

Le système prototype, organisé autour d'un microprocesseur 2650, génère quatre bus qui arrivent sur les connecteurs "F" (cf. fig. II.2.)

des différents microcalculateurs (fig. III.1.). Ces bus permettent d'accéder aux mémoires locales, aux registres des périphériques et à la mémoire commune.

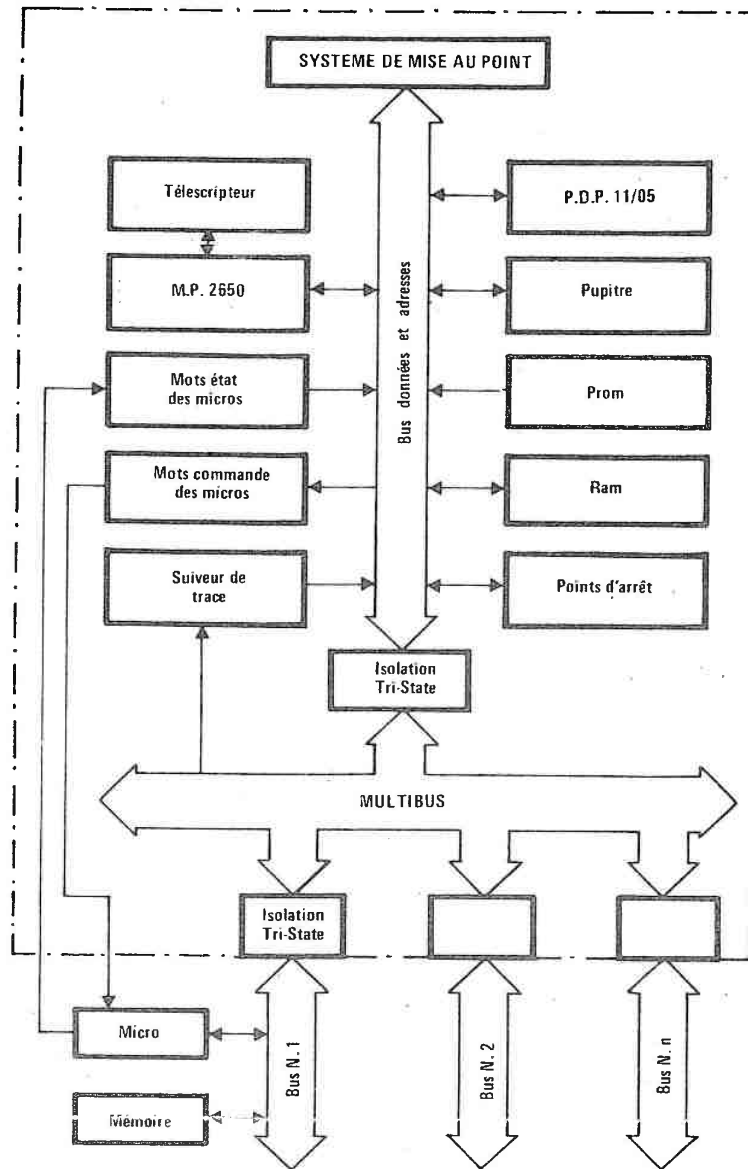


Figure III.1. : Organisation générale du système de mise au point.

Pour une plus grande souplesse d'utilisation, nous avons connecté ce système à un ordinateur PDP 11-05 équipé d'un disque, d'une imprimante rapide et d'un perforateur rapide qui permet de sortir un ruban objet assimilable par un programmeur de EPROM.

Le contrôle des microordinateurs se fait à partir d'un pupitre équipé d'une série de commutateurs, de LED et d'afficheurs hexadécimaux (fig. III.2.).

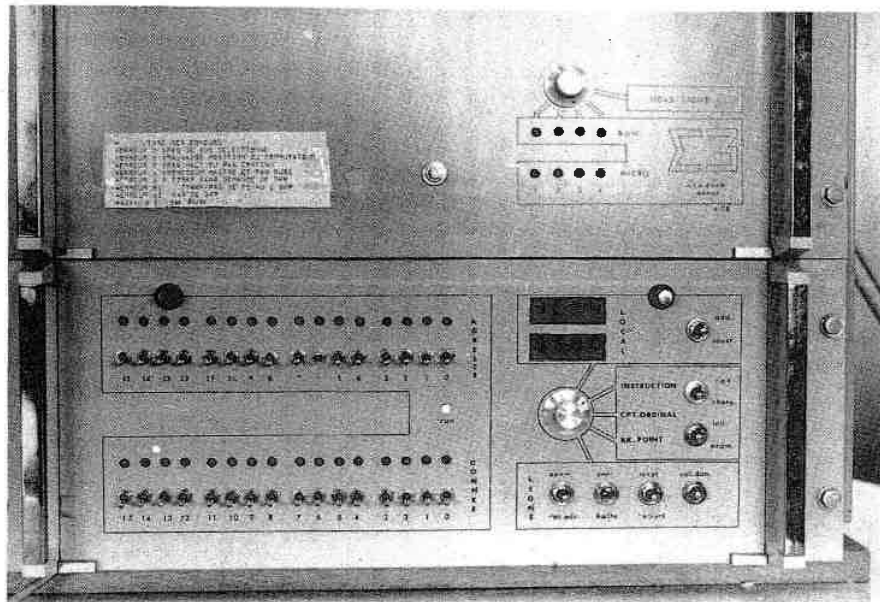


Figure III.2- Pupitre du système de mise au point

Ce pupitre permet le chargement, la modification des programmes en mémoire et le contrôle de leur exécution en temps réel.

Il est ainsi possible :

- de charger, à l'aide de commandes codées, un programme en mémoire vive volatile à partir du PDP 11 où il a été sauvegardé,
- de transférer une zone de la mémoire vive dans le PDP 11 où elle pourra être traitée, interprétée ou stockée,

- de lire ou de modifier un mot mémoire à partir des clés du pupitre,
- de lancer à une adresse quelconque un programme situé dans n'importe quel microcalculateur,
- de définir séparément pour chaque microprocesseur, un fonctionnement continu ou pas à pas,
- de fixer des points d'arrêt simultanément dans tous les microprocesseurs. On peut alors connaître la chronologie des arrivées aux différents points d'arrêt et visualiser ou modifier les registres et les mots d'état des différents processeurs arrêtés.

D'autre part, un suiveur de trace visualise les 256 dernières valeurs du compteur ordinal d'un microprocesseur quelconque sélectionné par commutateur. Cette possibilité est très utile dans la phase de mise en oeuvre pour connaître le déroulement réel du programme en présence d'interruptions multiples.

III-2-1) Signaux agissant sur les microprocesseurs contrôlés

Pour chaque microprocesseur contrôlé, le système développe un bus d'adresse de 14 bits, un bus de données de 8 bits, 4 signaux de service ($M/\overline{IOL}^*$, $\overline{M}/IOL^*$, $R/\overline{WL}^*$, $WRPL^*$), 3 signaux de commande ($RESET^*$, $PAUSE^*$, $OCCL^*$).

Ces signaux arrivent en parallèle sur les bus locaux de données et d'adresses et sur les collecteurs ouverts des signaux de service.

L'accès à la mémoire locale à partir du pupitre ne se conçoit que lorsque le microprocesseur concerné est en HALT et que ses bus sont à l'état "haute impédance". Le signal "BUS HZ", émis par le pupitre force les bus à prendre cet état et autorise l'accès à la mémoire et aux registres d'entrées-sorties en toutes circonstances (cf. fig. II.3.).

Le pupitre considère la mémoire locale d'un microcalculateur comme le prolongement de la sienne et le bit poids fort de l'adresse sert à différencier une adresse propre au pupitre, d'un transfert vers un microcalculateur.

III-2-2) Action matérielle sur le fonctionnement du microprocesseur

Une impulsion sur la ligne RESET* lance le microprocesseur à l'adresse 0, où un branchement permet d'atteindre le programme à exécuter.

Le signal PAUSE* arrête le processeur à la fin de l'instruction en cours et celui-ci se met en HALT.

Le fonctionnement pas à pas est obtenu en relachant le signal PAUSE* pendant un temps inférieur au temps d'exécution d'une instruction.

III-2-3) Action logicielle sur le fonctionnement du microprocesseur

Pour créer l'arrêt du déroulement d'un programme à une adresse donnée, il faut substituer le contenu de cette adresse par une instruction qui génère un signal à destination du pupitre afin que celui-ci puisse lancer la procédure d'arrêt.

Les instructions exécutées par le microprocesseur sont codées sur 1, 2 ou 3 octets ; si nous voulons mettre des points d'arrêt n'importe où et sans contrainte, il faut que l'instruction de substitution n'occupe qu'une seule place.

Nous avons choisi une instruction de sortie en mode non-étendu WRDT, codé sur un octet, et dont nous ne nous servons jamais pour adresser un périphérique.

La procédure permettant de créer un point d'arrêt est la suivante :

- remplacement à partir du pupitre du contenu de l'adresse d'arrêt par le code (FO) de l'instruction WRDT,
- lorsque le processeur décode cette instruction, le signal BKP* est généré (cf. fig. II.3.) et crée une interruption vers le pupitre,
- à ce moment, le pupitre met en attente le microcalculateur en le déclarant occupé (OCCL* = 0), ferme les bus locaux par le signal BUSHZ et insère, immédiatement après le code FO, une instruction de branchement à un programme spécifique de point d'arrêt,
- après ceci, le pupitre relache le signal OCCL* qui a pour effet de remettre en marche le processeur qui va alors se dérouter vers le programme de point d'arrêt,

- ce programme stocke la valeur des 7 registres et du mot d'état du processeur en mémoire vive, envoie une nouvelle interruption, par exécution de l'instruction WRDT, vers le pupitre qui le met alors en attente,
- le pupitre restitue la valeur originale des places mémoire modifiées et prépare l'adresse de retour en fin du programme de point d'arrêt,
- il est alors possible de lire et de modifier, à partir du pupitre, les valeurs des registres et du mot d'état rangés en mémoire,
- pour abandonner un point d'arrêt, le pupitre relache le signal occupé,
- alors le microprocesseur termine le programme de point d'arrêt qui restitue le contexte et va se brancher à l'adresse du point d'arrêt.

Il est évident que la possibilité des points d'arrêt nécessite le stockage du programme en mémoire vive.

Pour une version industrielle, nous remplacerons le pupitre de mise au point par un écran de visualisation équipée d'un clavier.

Le principe de fonctionnement restera le même, mais l'action sur les différents boutons sera remplacée par des commandes codées.

Cet équipement nous permettra de disposer d'un nombre d'informations plus important, aura une meilleure maniabilité et pourra être utilisé pour la maintenance du calculateur et le dépannage du processus industriel.

III-3) TEST ET MAINTENANCE DU CALCULATEUR

Le matériel utilisé par le pupitre de mise au point pour contrôler chaque microcalculateur permettra avec un programme adapté, de tester le bon fonctionnement de certains éléments d'une carte.

Un microcalculateur est composé de 5 sous-ensembles distincts qui sont :

- génération du bus local et du bus commun,
- génération des signaux de service,
- dialogue Moniteur-Esclave,

- mémoire locale,
- registre de commande des microcalculateurs esclaves,

et il est possible, en s'aidant du mot d'état du microcalculateur (cf. §.II-4-11), de localiser le sous-module défectueux.

La détermination de la connexion ou du circuit défaillant ne pourra se faire qu'avec un examen approfondi des différents signaux utilisés dans ce sous-module.

La façon la plus économique de procéder à cet examen est de demander à l'opérateur humain de sonder différents points en instaurant un dialogue entre l'opérateur et le système de maintenance par l'intermédiaire de la console de visualisation. Une recherche dichotomique doit aboutir à la localisation rapide de la panne.

III-4) DEPANNAGE DU PROCESSUS INDUSTRIEL

On peut imaginer de procéder de la même façon pour tester les différents organes d'un processus à l'aide d'un algorithme spécifique.

Par exemple, le système de dépannage propose une action sur un organe, attend l'autorisation de l'exécuter et demande le résultat à l'opérateur. Ainsi s'établit un dialogue qui assiste l'opérateur dans la recherche du diagnostic.

Une telle procédure qui semble discréditer l'opérateur humain au profit du système en lui donnant un rôle passif, est la seule façon, à notre avis, pour une personne ayant des compétences restreintes en électronique d'assurer la maintenance de son calculateur et de réduire les temps d'arrêt de son installation en cas de panne.

- 44 -

- 2ÈME PARTIE -

APPLICATION A UNE MACHINE
DE DECOUPE OPTIMALE DE PLANCHES

I N T R O D U C T I O N

Le travail que nous présentons maintenant, nous a été commandé par la Société MONTORNES spécialisée dans la construction de machines à bois. Il porte sur l'étude et la réalisation de la commande d'une machine de découpe optimale de planches.

Une première version avait été réalisée en mettant en oeuvre un calculateur MITRA 105 de la Société C.I.I. (4).

Le prix de revient et la nécessité d'utiliser un lecteur de disques souples en ambiance industrielle ont été les principaux obstacles à l'industrialisation de cette version.

Cette application rentrant exactement dans le créneau visé par notre calculateur, et fort de l'expérience acquise, nous avons repris l'étude afin d'en faire une réalisation adaptée aux besoins de la profession.

C H A P I T R E I

CAHIER DES CHARGES ET INTERFACES DE COUPLAGE

I-1) INTRODUCTION

A partir d'un stock de planches de diverses qualités et sections présentant des défauts repérés manuellement, la machine doit permettre d'obtenir, compte tenu d'un critère d'optimisation, des planches sans défauts dont la longueur est déterminée en fonction d'une production préalablement définie.

La commande d'un tel processus est caractérisée par les séquences suivantes :

- entrée des listes de découpes,
- mesure d'un produit,
- optimisation de la répartition des découpes dans les longueurs utiles, compte tenu d'un critère de chute minimale ou de prix de vente maximum,
- exécution des découpes,
- compte rendu de la production.

Nous allons préciser, dans les paragraphes suivants, le cahier des charges imposé par l'industriel et les moyens utilisés. Nous exposerons ensuite, comment nous avons mis en oeuvre notre calculateur pour conduire ce processus.

I-2) ENTREE DES LISTES DES DECOUPES

L'utilisateur spécifie sa production par une ou plusieurs listes de découpes qui sont définies par la qualité et la section du produit.

Chaque liste contenant les longueurs de planche désirées et le nombre à découper.

L'entrée de la production et la modification des listes sont effectuées à partir d'un écran de visualisation équipé d'un clavier sur un principe de question-réponse. Ce clavier sert également à définir certains paramètres de la machine (par exemple, épaisseur de la scie) et à éditer le compte rendu de la production.

Le nombre de listes et le nombre de longueurs dans chaque liste est fonction de la taille de la mémoire ; pour notre étude, nous nous sommes limités à deux listes de 15 longueurs.

I-3) MESURE D'UN PRODUIT

I-3-1) Détermination des longueurs utiles et de la qualité d'un produit

Pour garantir la rectitude de la section des planches, une chute est toujours prévue aux extrémités. Les autres zones inutilisables, présentant des défauts (noeuds, failles, corps étrangers), sont délimitées manuellement à la craie fluorescente lors de la préparation. Ces traits de craie sont détectés par une cellule à réflexion au moment de la mesure de la planche ; on connaît ainsi les portions de planches compatibles avec les exigences de la production.

L'interprétation des traits de craie obéit aux règles suivantes :

- un trait n'est pas pris en compte s'il se situe dans la zone de chute forcée aux extrémités (*figure I.1.a.*),
- la chute aux extrémités est augmentée jusqu'au trait si celui-ci se situe dans une zone définie par l'opérateur (longueur de rejet max. aux extrémités) (*figure I.1.b.*),
- en dehors de cette zone, un seul trait est interprété comme une frontière entre deux éléments utilisables à la condition que le prochain trait soit au-delà d'une certaine distance (chute max. admissible) ; ceci permet de minimiser la convexité de certaines planches (*figure I.1.c.*),
- les défauts ponctuels (noeuds isolés, par exemple) sont délimités par 2 traits de part et d'autre et dont l'espacement est inférieur à la chute maximale admissible (*figure I.1.d.*),
- une zone de défauts de longueur importante commence par trois traits proches et un trait unique en fin (*figure I.1.e.*).

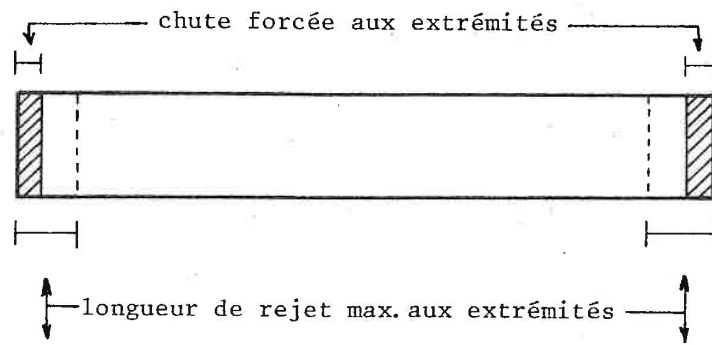


Figure I.1.a. : Chutes aux extrémités systématiques. (La partie utilisable est non hachurée).

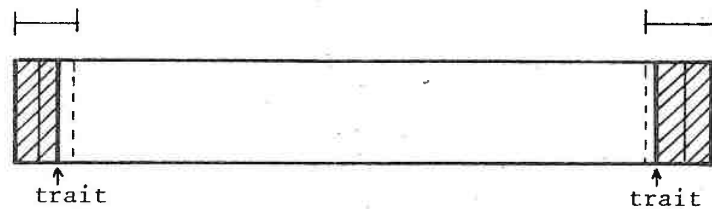


Figure I.1.b. : Chutes augmentées aux extrémités.

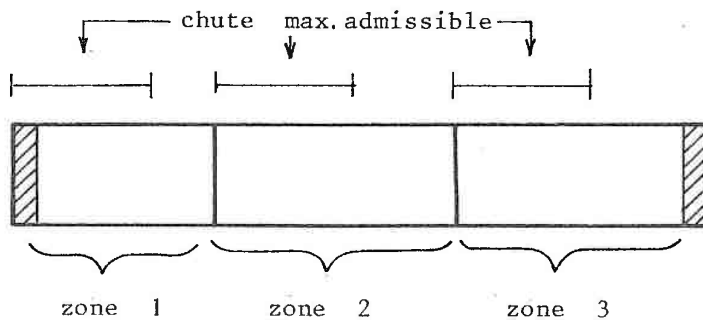


Figure I.1.c. : Trois zones utilisables délimitées par des traits.



Figure I.1.d. : Elimination des défauts ponctuels.

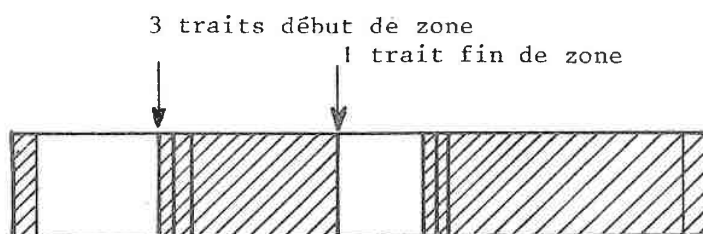


Figure I.1.e. : Elimination des défauts importants.

Figure I.1.

Lors de la création des listes de découpes, l'opérateur définit les différents paramètres nécessaires à la détermination des longueurs utiles :

- chute max. admissible,
- chute forcée aux extrémités,
- longueur de rejet max. aux extrémités,
- longueur max. entre trois traits.

I-3-2) Section et qualité d'un produit

La détermination de la section et de la qualité d'une planche peut se faire de deux façons :

- l'opérateur sélectionne, par une commande codée au clavier, la liste sur laquelle il veut travailler et qui correspond à la qualité et à la section du produit,

- la machine est équipée d'un système de mesure automatique de section et de qualité, les informations sont alors directement envoyées au calculateur. Dans ce cas, des planches de diverses sections et qualités peuvent être introduites indifféremment les unes derrière les autres.

La section est déterminée en mesurant les placements de palpeur venant en contact sur les faces de la planche.

La qualité d'une zone est codée manuellement sur le champ de la planche par 1, 2, 3 ou 4 traits de craie fluorescente.

Une zone qui ne présente pas de trait conserve la qualité précédemment prise en compte (*figure I.2.*).

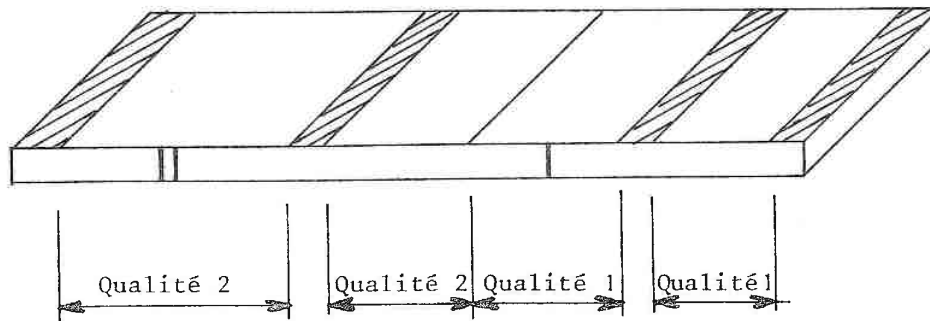


Figure I.2. : Codage de la qualité sur le champ de la planche.

I-3-3) Mesure de la planche

La mesure des planches est assurée par un générateur d'impulsions rotatif qui fournit deux impulsions déphasées de 90° pour un déplacement de 4 mm. Une multiplication par 2 de ces impulsions permet une résolution d'un millimètre (*figure I.3.*).

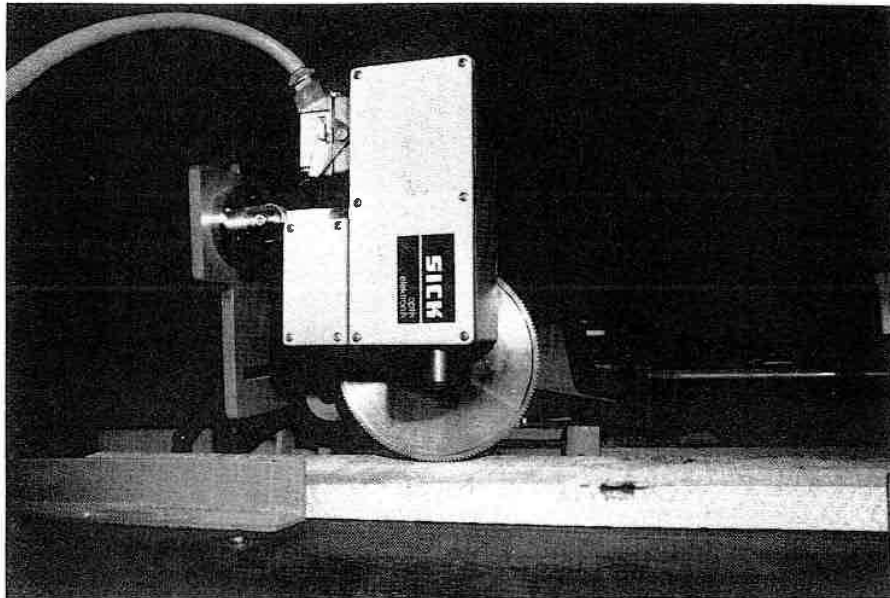


Figure I.3- Poste de mesure

En premier plan: Cellule de détection de trait défaut ,
derrière: disque solidaire du générateur d'impulsions.

I-4) DECOUPE DES PLANCHES

L'unité de découpe doit être capable de déplacer les planches à une vitesse maximale de 120 m/mn et de les positionner avec une précision de $\pm 0,2$ mm.

Ces contraintes nous ont amenés à utiliser, pour l'asservissement en position de la planche, un moteur à courant continu à rotor plat commandé par un variateur réversible à courant de circulation.

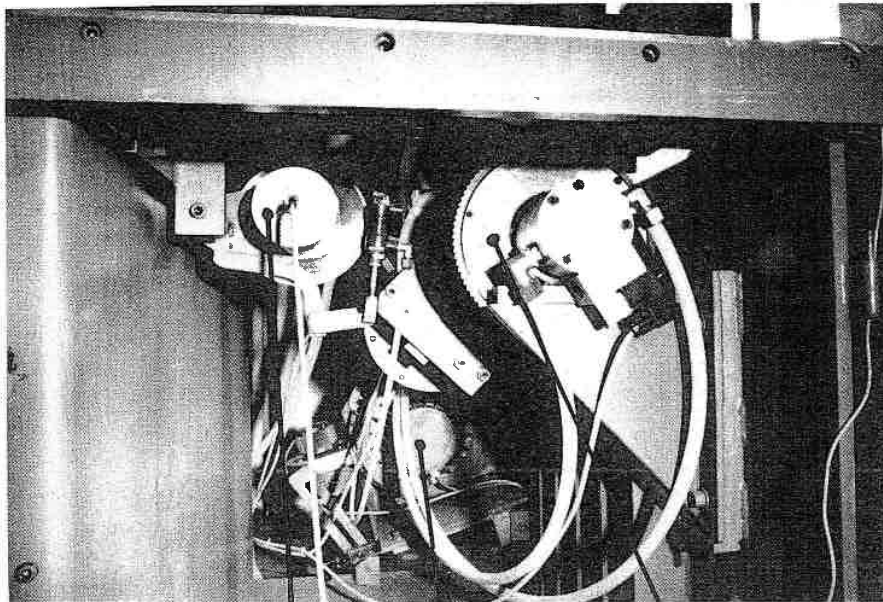
Cet ensemble a les caractéristiques suivantes :

- puissance normale : 5 KW,
- temps de passage de la vitesse maximale à l'arrêt : 400 ms,
- tension de commande minimale de démarrage : 5 mv,
- maintien d'un couple pour une tension de commande nulle ou déconnectée.

Un générateur d'impulsion rotatif, en contact avec la planche, permet de mesurer les déplacements de celle-ci pour la mettre en position.

D'autre part, l'unité de découpe comprend une scie circulaire mûe par un verin pneumatique et entraîné par un moteur asynchrone de 5,5 KW.

Des rouleaux presseurs assurent le maintien des planches et deux cellules photo-électriques, placées à l'entrée de l'unité de découpe, détectent les arrivées et les fins de planche (*figure I.4.*).



rouleau d'entraînement
moteur à courant continu
générateur d'impulsions

Figure I.4.a- Système d'entraînement et de mesure de la planche

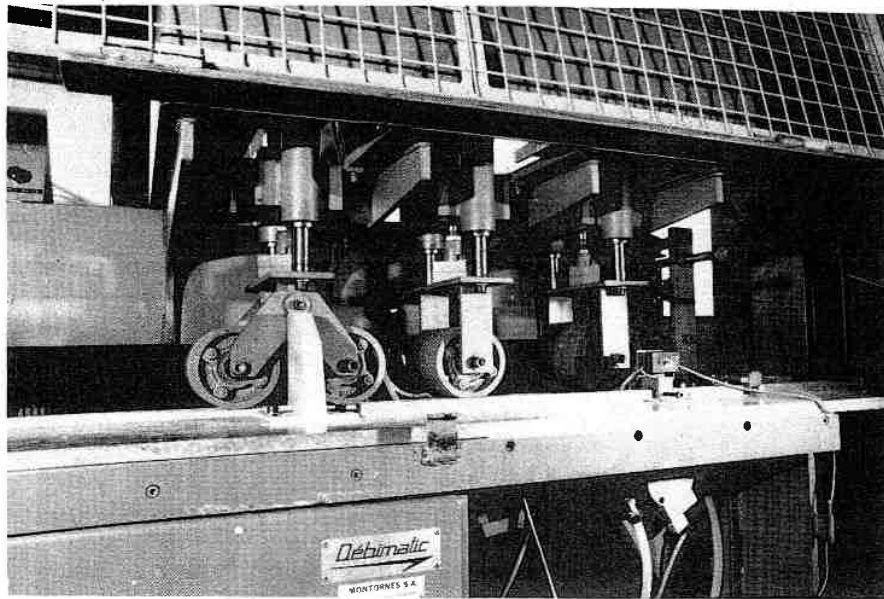


Figure I.4.b- Unité de découpe

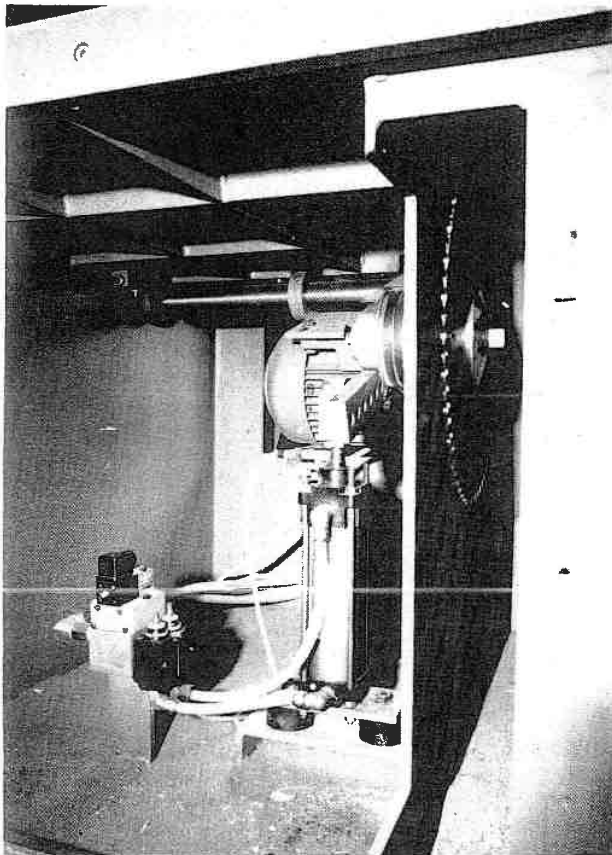


Figure I.4.c- Détail du système de sciage

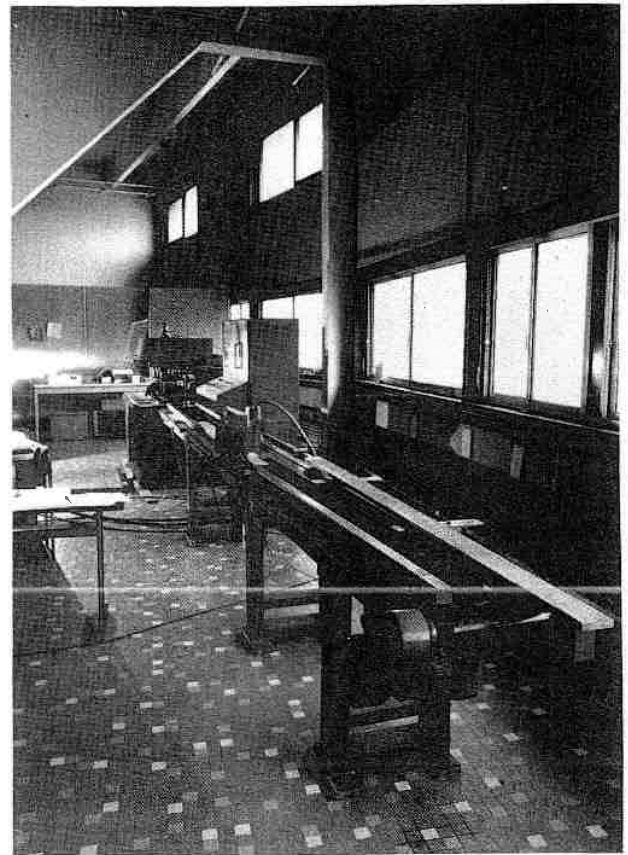


Figure I.4.d- Vue générale de la machine

I-5) ASSERVISSEMENT EN POSITION DE LA PLANCHE

L'asservissement en position de la planche fait appel à deux boucles de contre-réaction, l'une qui réalise l'asservissement en vitesse du moteur en utilisant une dynamo tachymétrique, l'autre qui assure le positionnement correct de la planche (*figure I.5.*).

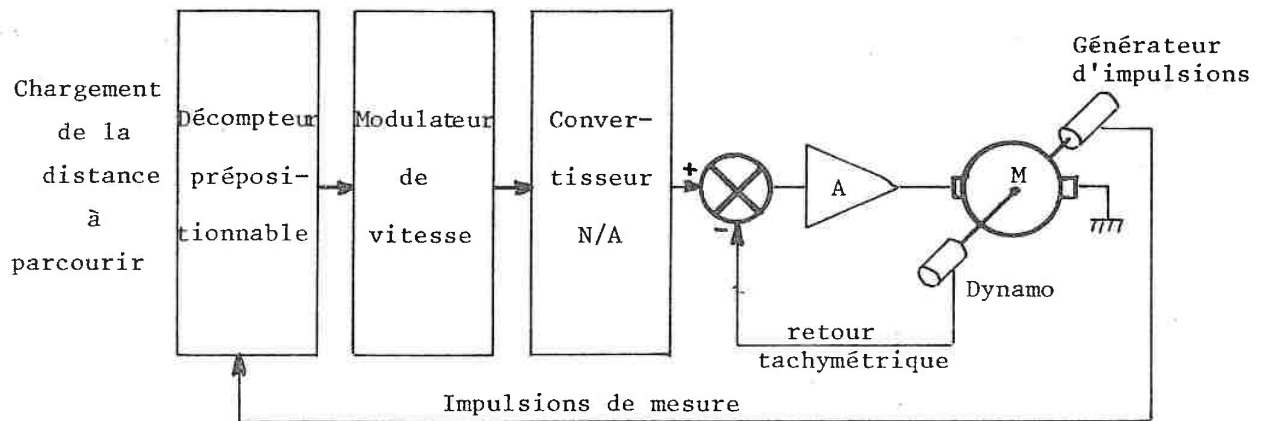


Figure I.5. : Schéma de l'asservissement en position des planches.

Nous avons étudié un montage permettant de réaliser cette seconde boucle de contre-réaction ; l'asservissement en vitesse étant intégré au variateur livré avec le moteur.

Le principe de fonctionnement est le suivant :

- le microcalculateur charge, dans un compteur-décompteur prépositionnable, un nombre d'impulsions représentant une certaine distance à parcourir,
- ce compteur est décrementé par les impulsions de retour émises par le générateur d'impulsions en contact avec la planche,

- la valeur de ce compteur est convertie, à tout instant, en une tension analogique qui est envoyée au variateur et qui impose une certaine vitesse au moteur.

Cette tension atteint un palier de saturation à partir d'une certaine valeur du compteur (*figure I.6.*).

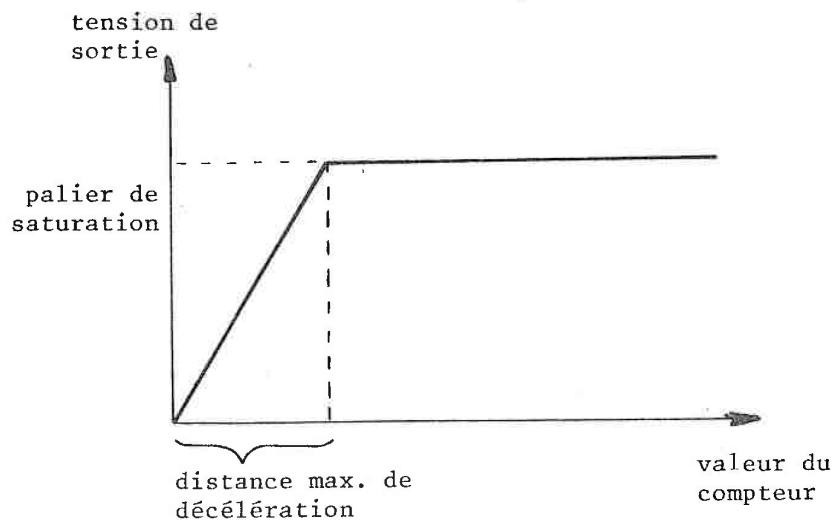


Figure I.6. : Tension de commande envoyée au variateur en fonction de la valeur du compteur.

Lorsque la distance restant à parcourir devient inférieure à cette valeur, la tension de commande diminue et une phase de décélération est ainsi générée. Durant cette phase, la tension de commande est proportionnelle à la distance restant à parcourir.

Pour assurer un positionnement sans dépassement, il faut que la valeur de saturation du compteur soit supérieure à la distance maximale nécessaire au moteur pour s'arrêter (*figure I.7.*).

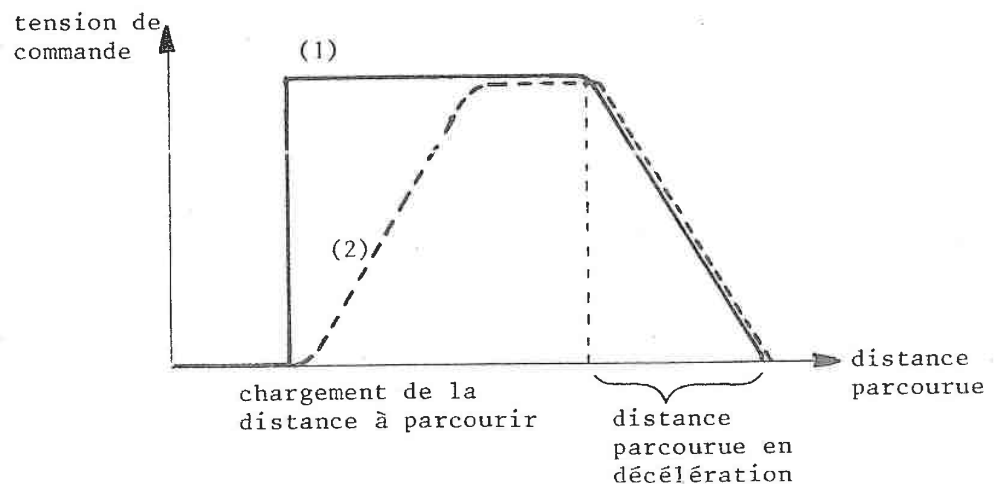


Figure I.7. : Courbe (1) : tension de commande,
Courbe (2) : tension tachymétrique.

Vu les caractéristiques dynamiques de ce moteur et les rapports de démultiplication, nous commençons à décélérer lorsqu'il ne reste plus qu'une distance de 409,6 mm à parcourir.

D'autre part, nous avons été obligé d'adjoindre à ce système un élément permettant de moduler la vitesse de déplacement des planches, sinon lorsqu'une planche quitte le rouleau d'entraînement en pleine vitesse elle continuerait à parcourir, sur son inertie, une distance incompatible avec les caractéristiques de la machine.

Une cellule, placée avant le rouleau, détecte la fin de planche et commande la mise en vitesse lente du moteur ; ainsi la planche quitte le rouleau d'entraînement à une vitesse acceptable.

Le modulateur de vitesse est composé de portes logiques qui masquent les 3 bits de poids forts envoyés au convertisseur numérique-analogique, on obtient donc quatre paliers de saturation différents commandables par le logiciel (*figure I.8.*).

Il est à noter que ce montage ne modifie en rien la pente de décélération générée.

Bit			Tension de Commande	Vitesse maximale
2^{12}	2^{11}	2^{10}		
1	1	1	10 V	3000 t/mn
0	1	1	5 V	1500 t/mn
0	0	1	2,5 V	750 t/mn
0	0	0	1,25 V	375 t/mn

Figure I.8. : Tableau des 4 paliers de vitesse maximale.

I-6) CARTES DE COUPLAGE CALCULATEUR-MACHINE

Nous avons également réalisé deux cartes de couplage, l'une affectée au poste de mesure, l'autre destinée à l'unité de découpe.

La carte "d'entrée-sortie mesure" comporte :

- un compteur de 16 bits qui comptabilise les impulsions émises par le générateur et qui peut être lu par le microcalculateur,

- un découplage opto-électronique pour les signaux provenant des cellules "présence planche" et de détection de trait, et les impulsions du générateur,

- une multiplication par 2 des impulsions de mesure.

La carte "d'entrée-sortie découpe" se compose :

- de plusieurs coupleurs opto-électroniques permettant de faire l'adaptation $+ 12\text{ V} \leftrightarrow + 5\text{ V}$ et d'isoler la calculateur des parasites extérieurs. Les signaux entrant proviennent des cellules :

- . présence planche à l'entrée de l'unité de découpe,
- . présence planche sur le rouleau d'entraînement,
- . scie basse.

Les signaux sortant commandent le presseur entrée, l'appui guide, la montée de la scie et le presseur scie,

- d'un système d'interruption qui envoie une interruption au microcalculateur lorsqu'une planche se présente à l'entrée de l'unité de découpe.

C H A P I T R E I I

ORGANISATION DU LOGICIEL

II-1) INTRODUCTION

Le processus que nous avons à commander fait apparaître quatre grandes tâches :

- une tâche de dialogue "homme-machine" qui permet de définir les listes de découpes de planches, et qui édite les résultats de la production ;
- une tâche de mesure qui détermine les caractéristiques géométriques et les zones utilisables de la planche ;
- une tâche d'optimisation qui recherche la découpe optimale du produit suivant un critère de prix de vente maximum des éléments coupés ;
- une tâche de découpe qui assure la découpe des éléments définis par l'optimisation.

Ces tâches sont séquentiellement coopérantes pour une planche, mais la présence de plusieurs planches entre le poste de mesure et le poste de découpe nécessite de mémoriser les informations et de les exploiter suivant les besoins de la machine.

L'utilisation d'un ordinateur uniprocésseur oblige le concepteur à définir une priorité de traitement, ce qui pénalise certaines tâches.

Or, il est évident que l'optimisation sera d'autant plus performante qu'elle disposera d'un maximum de temps de recherche.

D'autre part, la précision de découpe est conditionnée par la prise en compte de l'interruption de fin de planche qui permet de recadrer la planche suivante, afin d'éviter les erreurs cumulatives.

Ces deux points justifient, à eux seuls, l'emploi d'un ordinateur multiprocesseurs.

Nous avons donc décidé d'affecter un micro-ordinateur à la tâche d'optimisation, et un micro-ordinateur à la tâche de découpe.

Nous aurions pu imposer de ne permettre le dialogue "homme-machine" que lorsque la machine est à l'arrêt, et de rendre ainsi la tâche dialogue indépendante. Nous avons cependant préféré affecter un micro-ordinateur à cette tâche, afin de faciliter ultérieurement la communication de notre système avec un ordinateur placé en amont.

L'affectation d'un micro-ordinateur à la tâche de mesure permet un découpage du système correspondant aux différentes unités de la machine, ce qui doit faciliter la maintenance.

Le ordinateur mis en oeuvre se compose donc de quatre modules micro-ordinateurs "ESCLAVE", d'un module "MONITEUR" qui assure la synchronisation des différentes tâches, et d'une mémoire commune servant de boîte à lettre (*figure II.1.*).

Nous présentons dans les paragraphes suivants l'organisation de la mémoire commune et les principes utilisés pour l'exécution des tâches dialogue, mesure, optimisation et découpe.

Le fonctionnement précis de chacune de ces tâches et le détail de l'analyse sont présentés en annexe technique.

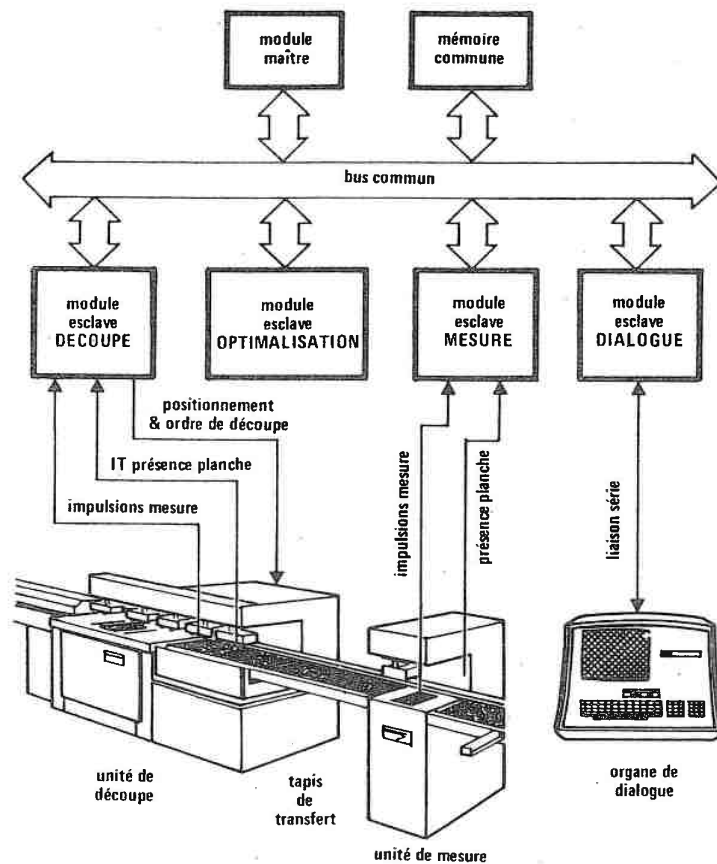


Figure II.1- Organisation générale du calculateur mis en oeuvre

II-2) ORGANISATION DE LA MEMOIRE COMMUNE

Nous avons créé quatre files en mémoire commune pour le transfert des données d'un micro-calculateur vers un autre (*figure II.2.*).

Ces quatre files sont :

- TLM (Tableau des Longueurs Marchandes) qui contient la liste des découpes par section et par qualité, et qui est chargée par la tâche DIALOGUE. La tâche OPTIMISATION utilise cette file pour connaître les découpes demandées et pour mettre à jour le nombre de longueurs restant à découper.

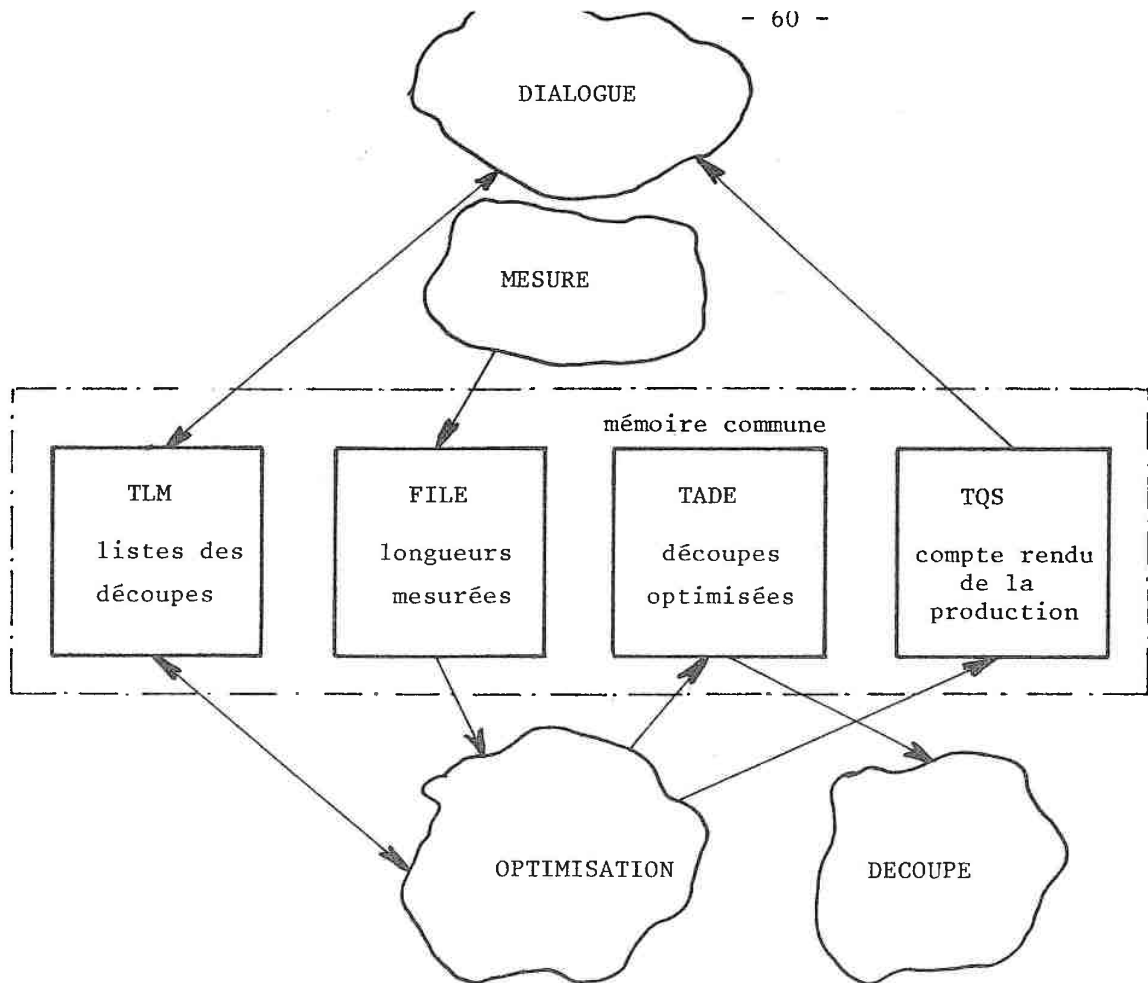
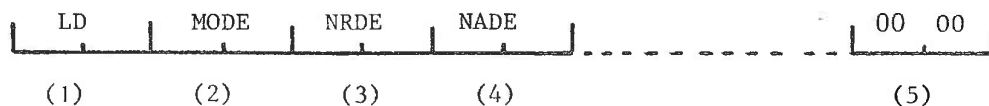


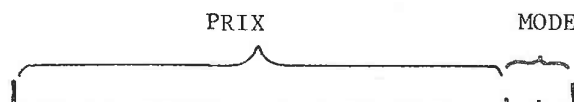
Figure II.2

Le format de stockage est le suivant :



(1) Longueur à découper en 0,1 mm sur 16 bits.

(2) Codage du prix sur 14 bits et du mode sur 2 bits. Le mode permet de forcer une découpe dans une longueur utile sans tenir compte du critère d'optimisation ou d'abandonner temporairement la découpe d'une longueur.



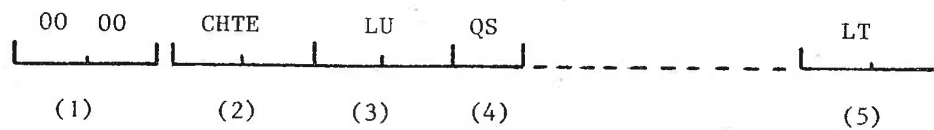
0 0 - pas de forçage-normal
 0 1 - forçage
 1 0 - suppression

- (3) Nombre de longueurs restant à découper.
- (4) Nombre de longueurs demandées.
- (5) Indicateur de fin de tableau.

- FILE (file des longueurs mesurées) qui est chargée par la tâche mesure à chaque longueur utile mesurée, et qui est utilisée par la tâche optimisation.

Une longueur utile suit toujours une chute puisque nous forçons systématiquement une chute en début de planche.

La disposition dans cette file circulaire est :



- (1) Indicateur de début de planche.
- (2) Longueur de la chute.
- (3) Longueur utile.
- (4) Codage de la qualité et de la section.
- (5) Longueur totale de la planche.

- TADE (tableau des découpes) qui est le résultat de l'optimisation et qui constitue la ressource de la tâche DECOUPE. Cette file circulaire est constituée par une suite de longueurs à découper avec un indicateur de début de planche, afin de permettre le recadrage en début de chaque planche.

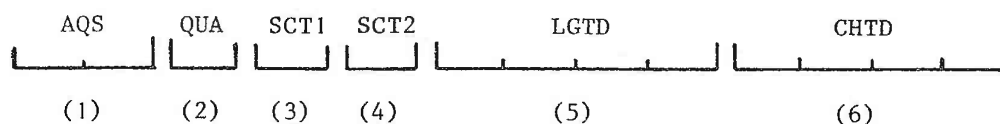


- (1) Indicateur de début de planche.
- (2) Longueur à découper.

- TQS (tableau des qualités-section) qui contient la liste des diverses qualité-section de planches utilisées. Ce tableau est exploité par l'optimisation pour :

- déterminer l'adresse en mémoire du tableau TLM relatif à la qualité-section de la longueur utile à optimiser ;
- gérer la longueur totale utile de planches et la longueur totale des chutes. Ces informations sont éditables grâce au DIALOGUE et donnent à l'utilisateur une indication du rendement de son produit.

Les éléments dans ce tableau sont :



(1) Adresse de début du tableau TLM.

(2) Qualité.

(3) et (4) Section du produit en mm.

(5) Longueur totale utile.

(6) Longueur totale des chutes.

II-3) TÂCHE DIALOGUE

La tâche DIALOGUE qui assure la communication homme-machine, est basée sur le principe Question-Réponse.

Ainsi, à la mise sous tension, le MONITEUR active la tâche DIALOGUE qui teste le contenu d'une adresse en mémoire commune. Si cette adresse ne contient pas une valeur prédéterminée, cela signifie que la mémoire commune est détruite et que les informations relatives aux paramètres de la machine et aux tableaux de découpes sont erronées ; c'est ce qui se passe lorsque la mémoire commune est réalisée en mémoire volatile non sauvegardée.

Dans ce cas, la tâche DIALOGUE demande à l'opérateur les renseignements suivants :

***** PARAMETRES *****

CHUTE MAX. ADMISSIBLE	?	X X X X
CHUTE FORCEE AUX EXTREMITES	?	X X X X
LONG DE REJET MAX AUX EXTREMITES	?	X X X X
EPAISSEUR SCIE	?	X X X X
HAUTEUR DE SCIAGE	?	X X X X
LONGUEUR MAX ENTRE TROIS TRAITS	?	X X X X
COEFFICIENT CODEUR MESURE	?	X X X X
COEFFICIENT CODEUR DECOUPE	?	X X X X
LONGUEUR SCIE CELLULE	?	X X X X
TEMPO APPUI PRESSEUR	?	X X X X

Les longueurs sont en 1/10 de millimètre et la temporisation en 1/10 de seconde.



Figure II.3- Console de communication

Comme il n'y a aucun tableau de découpes en mémoire, la tâche DIALOGUE impose à l'opérateur de créer un tableau et affiche :

*** TABLEAU DES LONGUEURS A DECOUPER ***

QUALITE : X X X

SECTION : X X X / X X X

<u>LONGUEUR</u>	<u>PRIX</u>	<u>QUANTITE</u>	<u>MODE</u>	<u>QUANTITE NON DECOUPEE</u>
X X X X	X X	X X	X X	X X
.	.	.	.	.
.	.	.	.	.
X X X X	X X	X X	X X	X X

L'opérateur peut définir 15 longueurs au maximum.

Les longueurs exprimées en 0,1 mm ne peuvent excéder 6,5 m.

Le prix sert pour le critère d'optimisation. Si les prix des différentes longueurs sont proportionnels, l'optimisation se fait sur un critère de minimisation de la chute ; si au contraire on affecte un prix linéaire plus important aux grandes découpes, alors on obtient une optimisation du prix de vente.

Les colonnes QUANTITE et QUANTITE NON DECOUPEE spécifient le nombre de découpes demandé.

Le MODE permet de forcer une découpe (valeur 1) ou de l'abandonner (valeur 2) et la valeur 0 correspond à un fonctionnement normal.

Lorsqu'un tableau de découpe est défini, la tâche DIALOGUE le signale au MONITEUR qui lance les micro-calculateurs de MESURE, d'OPTIMISATION et de DECOUPE. La machine est prête à fonctionner.

III-3-1) Commandes codées :

Nous avons prévu des commandes codées qui permettent de travailler sur les paramètres et les tableaux de découpes.

Ces commandes sont :

- VPAR, qui visualise tous les paramètres précédemment définis ;
- MPAR, qui permet la modification des paramètres ;
- QSEC, qui permet de changer de tableau en définissant un nouveau couple Qualité-Section ;
- VTBL, qui visualise le tableau Qualité-Section relatif au QSEC précédemment défini ;
- ATBL, qui annule le tableau de découpes relatif à une Qualité-Section ;
- CTBL, qui assure la création d'un tableau de découpe ;
- MLON, qui permet de modifier une longueur d'un tableau ou d'insérer des longueurs nouvelles en fin de tableau ;
- LIST, qui rappelle à l'opérateur les tableaux de découpes actuellement en mémoire et qui indique, par Qualité-Section, le total des longueurs utiles découpées et le total des longueurs de chute, y compris l'épaisseur scie ;
- ALCT, qui remet à zéro ces deux totalisateurs ;
- VISM, qui permet de visualiser une série de places mémoire. Cette commande a été introduite pour faciliter la maintenance du système.

Toutes les commandes qui ne font que des lectures en mémoire commune peuvent être utilisées simultanément à la marche de la machine. Par contre, toute commande nécessitant une écriture en mémoire commune, ne sera prise en compte que lorsque l'optimisation et la découpe seront en attente. Le moniteur relancera alors le dialogue et bloquera la marche de la machine pour éviter, en particulier, que l'optimisation ne travaille sur des tableaux de découpes en cours de modification.

La commande EXEC permettra ensuite de remettre en route la machine, et précisera quel est le tableau de découpe qui doit être utilisé par l'optimisation (ceci

uniquement si la machine ne dispose pas de dispositif de mesure automatique de la Qualite-Section de la planche).

Les commandes destinées à effectuer des modifications (MPAR, MLON, QSEC) affichent successivement la valeur des différentes grandeurs concernées. Si l'on désire modifier une valeur, il suffit de frapper celle-ci au clavier. Si, par contre, on veut conserver la valeur précédente, il faut frapper un ESPACE qui fait évoluer l'affichage.

Le caractère LINE-FEED permet d'abandonner la commande et toutes les grandeurs qui n'ont pas été affichées gardent alors leur ancienne valeur.

II-3-2) Interpréteur de commandes :

L'exécution de ces commandes codées demande à la tâche DIALOGUE :

- de sortir des phrases et des valeurs numériques ;
- de contrôler les valeurs entrées au clavier ;
- d'attendre, dans certains cas, des réponses venant de l'opérateur ;
- de répéter certaines questions.

Pour satisfaire à ces exigences et pour disposer d'une grande souplesse d'adaptation afin d'autoriser de nouvelles commandes, nous avons mis au point un interpréteur de commandes. Chaque commande est donc codée sous forme d'un tableau précisant le format de celle-ci.

Le tableau de codage d'une commande est composé de plusieurs sections dont la signification est précisée par un mode.

Il existe six modes différents spécifiques à l'édition de texte, à l'attente de valeur, à l'appel de sous-programmes, à la recherche dans des tableaux de valeurs.

Chaque section est composée de divers paramètres utiles à l'interprétation du mode correspondant. Par exemple, pour le mode d'affichage d'une valeur numérique, la section est composée des paramètres suivants :

- les numéros des textes à visualiser ;
- l'adresse de la variable dont on veut afficher la valeur ;
- le format de cette variable.

D'autre part, un chaînage en début de tableau permet au système d'effectuer une recherche rapide du tableau relatif à la commande frappée au clavier.

Ainsi, par exemple, pour la commande codée VTBL permettant de visualiser un tableau de découpes relatif à une qualité et à une section, nous obtenons le tableau de traitement suivant :

VTBL	DATA	A 'VTBL'	(1)	
	DATA	H '11'	(2)	
	DATA	250	(3)	} Première section
	DATA	D'2, 13, 30, 0'	(4)	
	DATA	253	(5)	} Deuxième section
	ACON	QSE1	(6)	
	DATA	253	(7)	} Troisième section
	ACON	CTB1	(8)	
	DATA	252	(9)	Fin de commande

- (1) Mot clé de 4 caractères identifiant le tableau de traitement de la commande VTBL.
- (2) Chainage permettant d'atteindre le prochain tableau de traitement.
- (3) Mode 250 qui spécifie que le paramètre suivant est un texte à éditer sans répétition.
- (4) Liste des numéros des textes à éditer :
 - 2 (x x x x x x x x),
 - 13 (TABLEAU DES LONGUEURS A DECOUPER),
 - 30 (retour chariot, line feed),
 - et 0 qui indique la fin de la liste des textes.

- (5) Mode 253 qui indique que l'on veut utiliser le format d'une autre commande.
- (6) Ainsi, en exploitant le tableau de traitement relatif à la commande QSEC (visualisation de la Qualité-Section courante) à partir de l'étiquette QSEI on visualise le texte suivant :

QUANTITE : X X X SECTION : X X / X X .

- (7) (8) Mode 253 qui utilise une partie du tableau de traitement de la commande CTBL (création d'un tableau de longueurs à découper) à partir de l'étiquette CTBI pour afficher les 15 longueurs du tableau de découpe relatif à une Qualité-Section.

Ce mode permet donc de faire des branchements dans d'autres tableaux de traitement et de revenir en séquence en fin de tableau lorsqu'on rencontre le mode 252.

- (9) Mode 252 indiquant la fin du tableau de traitement de la commande VTBL.

Cet interpréteur est détaillé en annexe technique, ainsi que les différents paramètres utilisés.

II-4) TÂCHE MESURE

Lorsqu'une planche commence à être mesurée, le programme de fond de la tâche MESURE consiste en une scrutation cyclique de l'état des cellules de détection de trait défaut, de trait qualité et de présence planche.

Un trait défaut ne peut être interprété que lorsque l'on connaît la position des prochains traits ou de la fin de planche.

Nous avons donc utilisé un automate dans lequel on progresse à chaque détection de trait (cf. annexe technique) ; la distance entre chaque trait déterminant le cheminement.

Lorsqu'une longueur utile est trouvée, la tâche MESURE range la chute précédente, la longueur utile et le couple Qualité-Section (QS) dans le tableau FILE, et envoie une interruption vers le module MONITEUR pour lui signaler qu'une longueur utile peut être optimisée.

En fin de planche, la tâche range dans FILE le dernier élément chute-longueur utile-QS et la longueur totale de la planche.

II-5) TÂCHE OPTIMISATION

Le critère d'optimisation fait intervenir deux grandeurs :

- la chute maximale admissible au-dessus de laquelle la planche n'est pas utilisée,
- le prix de chaque élément de découpe.

Le programme d'optimisation cherche donc à descendre en-dessous de cette chute et recherche la combinaison des découpes qui permet d'obtenir le meilleur prix de vente.

Généralement, le prix de vente linéaire du bois n'est pas constant, mais augmente avec la longueur. L'affectation d'un prix à chaque planche permet de tenir compte de ce facteur, et l'optimisation sera alors effectuée selon un critère de prix de vente maximal.

Par contre, si l'utilisateur désire minimiser la chute, il lui suffira d'afficher pour chaque planche un prix strictement proportionnel à la longueur de celle-ci ; le critère de prix de vente maximal assurera alors une chute minimale.

Etant donné que nous ne disposons pas d'arithmétique câblée et que nous travaillons sur deux octets, la recherche exhaustive des possibilités de découpe pour 15 longueurs serait incompatible avec le temps de déplacement de la planche entre le poste de mesure et le poste de découpe.

D'autre part, il paraît logique d'accorder un temps de recherche supérieur aux grandes longueurs qu'aux petites longueurs.

Nous avons donc réalisé une horloge programmable qui est initialisée, en début d'optimisation, à une valeur tenant compte de la longueur utile à optimiser et de la vitesse de déplacement du tapis. Lorsque le temps imparti est écoulé, l'horloge envoie une interruption au module OPTIMISATION qui arrête sa recherche.

II-5-1) Programme d'optimisation :

Au démarrage de la machine, la tâche OPTIMISATION initialise certaines variables et se met en attente en exécutant l'instruction HALT, puisqu'aucune longueur utile n'est encore mesurée.

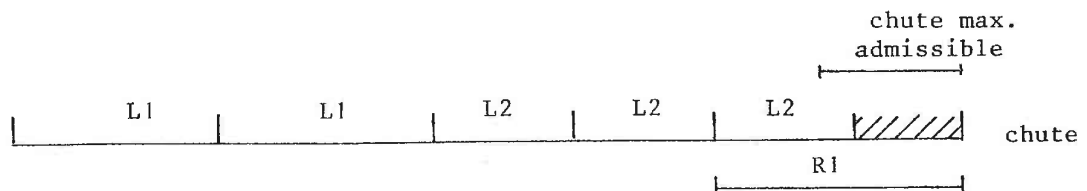
Aussitôt qu'une longueur utile est disponible dans le tableau FILE, le moniteur envoie une interruption à destination du module optimisation qui provoque la réactivation de la tâche OPTIMISATION.

Les séquences de traitement sont alors les suivantes :

- recherche de la chute, de la longueur utile et de la Qualite-Section dans FILE ;
- recherche, dans le tableau TLM, des longueurs qui peuvent être découpées dans cette longueur, du nombre restant à découper, du prix et chargement dans un tableau en mémoire locale ;
- initialisation de l'horloge programmable ;
- entrée dans le boucle d'optimisation.

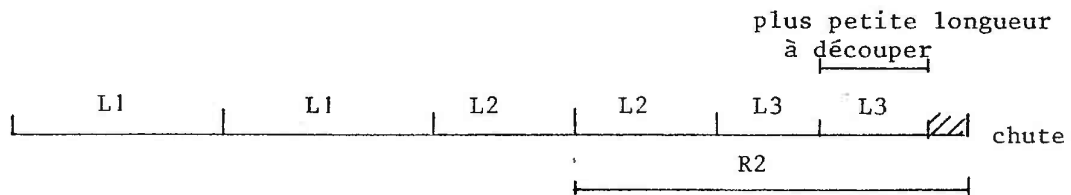
Comme le critère d'optimisation est le prix, on essaye de placer les plus grandes découpes et, à l'issue du premier tour, on dispose d'une première combinaison de longueurs qui correspond à un certain prix de vente.

Ce résultat est bon si la chute restante est inférieure à la chute maximale admissible.



Résultat après le premier tour dans la boucle d'optimisation

Ensuite, on décrémente de 1 le nombre des plus petites longueurs précédemment trouvées (L2) et on retourne dans la boucle d'optimisation afin d'optimiser cette longueur augmentée de la chute, (R1).



Résultat après le deuxième tour dans la boucle d'optimisation.

Si au deuxième tour le prix de vente est meilleur, on conserve cette solution.

Lorsque la plus petite longueur trouvée est la plus petite longueur à découper, alors on ôte la longueur immédiatement précédente (L2) et on fait une recherche sur la longueur R2, et ainsi de suite.

Seule la combinaison donnant le meilleur prix de vente est prise en compte.

La recherche se termine au moment où ne sont placées dans la longueur utile que les plus petites longueurs à découper, mais il est probable que cette recherche soit interrompue par l'horloge.

Lorsque l'optimisation est terminée, le programme charge le résultat dans le tableau TADE, met à jour le nombre de longueur restant à découper dans le tableau TLM, et actualise la longueur totale utile et la longueur totale des chutes dans le tableau TQS.

Ces opérations terminées, l'optimisation positionne un indicateur en mémoire commune pour signaler qu'elle est disponible, envoie une interruption vers le moniteur et se met en HALT.

Si une nouvelle longueur utile est à optimiser, le moniteur réactive la tâche OPTIMISATION en lui envoyant une interruption.

II-6) TÂCHE DE DECOUPE

L'entrainement de la planche est assuré par un rouleau couplé à un moteur à courant continu. La distance à parcourir est chargée dans un compteur-décompteur prépositionnable qui est décrémenté par les impulsions venant du générateur en contact avec la planche. Cette valeur (DIS) est, à tout instant, traduite en une tension analogique qui est envoyée au variateur pilotant le moteur (cf. I-5).

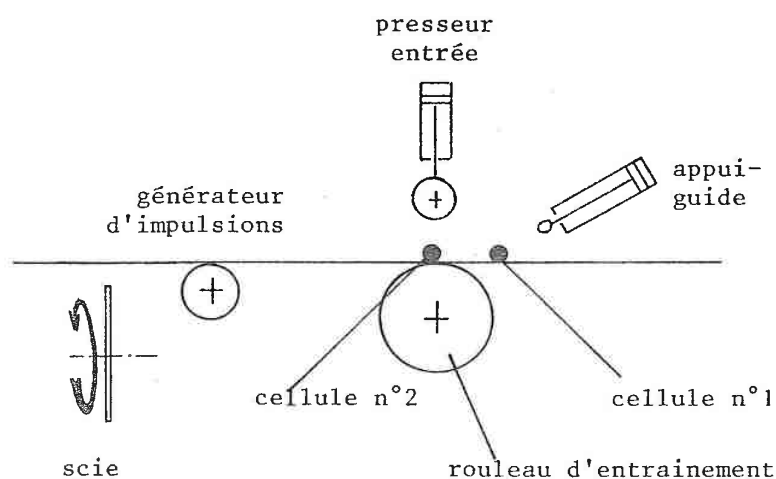


Figure II-3

La planche entraînée poussant la précédente, il est nécessaire, afin de ne pas avoir d'erreurs de positionnement cumulatives, de recadrer chaque fin de planche.

La figure II-4 montre, sur un exemple, la manière employée pour recadrer une planche.

Une cellule n° 1 détecte la fin d'une planche et envoie une interruption au module découpe. Comme la planche se déplace, la valeur (DIS1) du compteur correspond à la distance restant à parcourir pour arriver jusqu'à la prochaine découpe. Si l'on connaît précisément la distance (LSCL) entre la cellule n° 1 et la scie, on est capable de déterminer la valeur REST qui est égale à $LSCL - DIS1$ (figure II-4-a).

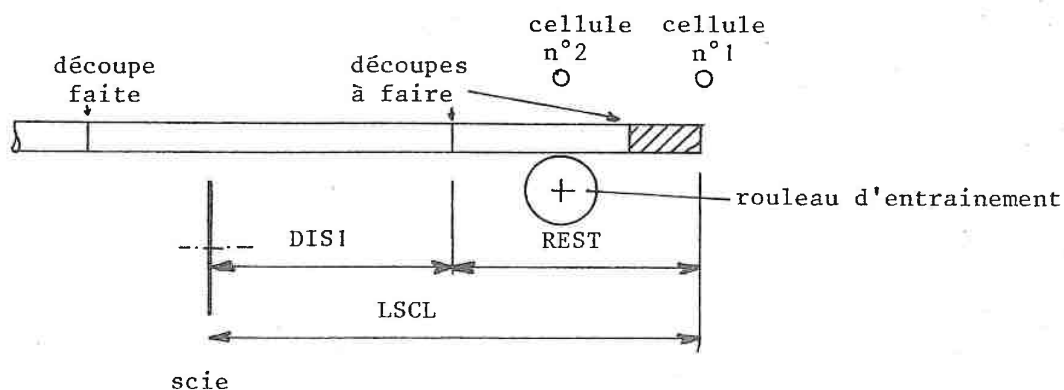


Figure II-4-a

La planche continue à se déplacer et quitte le rouleau d'entraînement. La cellule n° 2 le signale à la tâche DECOUPE qui relève le presseur "entrée" et dégage l'"appui-guide".

L'arrivée d'une nouvelle planche est détectée par la cellule n° 2. On pousse alors la planche sur le guide en commandant le vérin "appui-guide" et on abaisse le presseur "entrée".

A ce moment, la planche est entraînée et vient pousser la planche précédente pour l'amener en position de découpe (figure II-4-b).

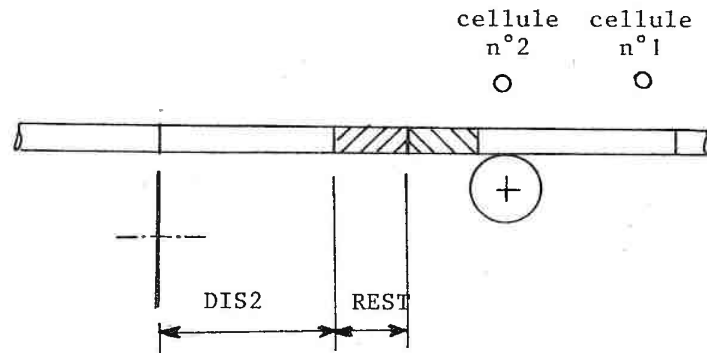


Figure II-4-b

On détermine $REST = REST - DIS2$ et on charge $DIS2$ dans le compteur.

Après le déplacement demandé, il ne reste plus de longueur à découper dans la première planche, et le début de la deuxième planche est une chute ; la prochaine découpe doit donc être à la distance :

$$DIS3 = REST + CHUT$$

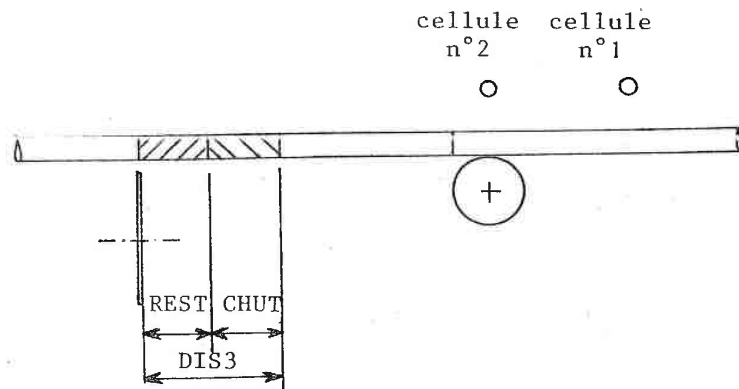


Figure II-4-c

C'est cette valeur que l'on charge dans le compteur et qui permet de venir scier la chute de début de la nouvelle planche.

Plusieurs petites planches pouvant être engagées entre la cellule n° 1 et la scie, il est nécessaire de gérer plusieurs restes. Nous avons donc prévu un tableau qui permette de mémoriser 5 restes au maximum.

II-6-1) Programme de découpe :

Le logiciel de la découpe est composé d'un programme de fond pour charger le compteur, et d'un sous-programme d'interruption qui gère l'interruption issue de la cellule n° 1 pour assurer le cadrage des planches.

Le programme de fond est le suivant (figure II-5) :

- recherche d'une longueur à découper dans le tableau TADE, lorsque la longueur précédente a été sciée ;
- test de la valeur de ce compteur ;
- commande de sciage lorsque le compteur est à zéro.

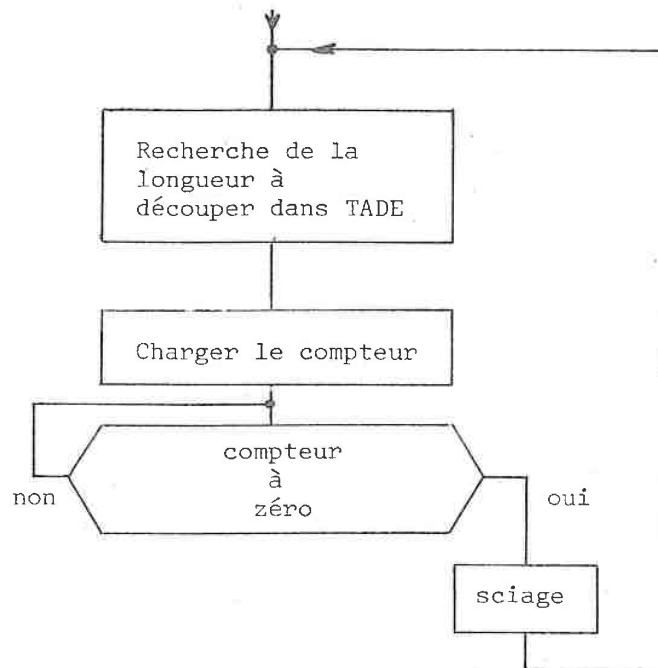


Figure II-5 : Programme de fond de la tâche DECOUPE

Le sous-programme d'interruption assure la lecture du compteur pour la gestion des restes et le contrôle des successions logiques de début et de fin de planche.

II-7) MONITEUR

Le programme assurant la coordination des différentes tâches se compose d'une séquence d'initialisation et d'un sous-programme d'interruption.

Les interruptions sont générées par les divers modules esclaves lorsque ceux-ci ont un évènement à signaler au moniteur. La nature de l'évènement est définie par un indicateur placé en mémoire commune (cf. II-4-8).

Cet indicateur est testé par le Moniteur lorsqu'il reçoit une interruption, ce qui lui permet de générer les transitions correspondantes, par exemple :

- relance de l'optimisation lorsque la mesure a déposé une nouvelle longueur utile dans FILE,
- relance de la découpe lorsque l'optimisation a fourni une liste de découpes dans TADE.

L'ensemble des transitions est présenté en annexe technique.

C O N C L U S I O N

Les possibilités et la modularité inhérentes aux calculateurs multiprocesseurs les appellent à un développement dans les prochaines années.

Dans le domaine de la commande de machines spéciales par calculateur, nous avons essayé de définir les performances matérielles et logicielles demandées au calculateur, et de limiter le matériel au minimum.

Parallèlement à la réalisation de ce calculateur, nous avons mis en oeuvre un outil de développement permettant l'accès simultané aux différents processeurs.

A notre connaissance, aucun constructeur de microprocesseurs ne propose un tel système qui est indispensable pour une mise en oeuvre rapide de programmes et une maintenance aisée.

En effet, nous pensons que l'utilisation du calculateur dans un nombre grandissant d'applications doit s'accompagner de la notion de "self-depannage" qui doit permettre d'effacer certaines hésitations d'industriels devant le prix demandé pour la maintenance de leur calculateur. Dans ce sens également, notre système présente une certaine originalité.

D'autre part, il est à noter que le prix de revient d'une carte microcalculateur complète est d'environ 1 000 F. et que la réalisation du calculateur (avec ses cartes d'entrée-sortie) nécessaires à la commande de la machine de découpe optimale de planches revient à environ 12 000 F. ; soit le prix des cartes d'entrée-sortie achetées pour le couplage avec le MITRA 105 lors de la précédente réalisation.

Enfin, les résultats obtenus dans la conduite de la machine à bois ont confirmé la validité des options prises au moment de la conception. Ainsi, la modularité logicielle et matérielle autorise l'adjonction d'un module microcalculateur destiné à commander le tri des planches découpées, et cela sans toucher aux autres modules, ni affecter les caractéristiques dynamiques de la machine.

B I B L I O G R A P H I E

- (1) "L'automatisation par l'intelligence répartie", S. KAHNE, I. LEFKOWITZ, C. ROSE, *Pour la Science*, aout 1979.
- (2) "Système modulaire multitâches pour commande de processus", J.-P. MUSSE, J.-P. DRAPIER, O. DOUCHIN, *Nouvel Automatismes*, sept.-oct. 1979.
- (3) "Conception et réalisation d'un multicalculateur temps réel à grande sûreté de fonctionnement", P. VERNEL, Thèse d'Etat, I.N.P.L. Nancy, avril 1977.
- (4) "Réalisation d'un système de découpe optimale de planches :
- mise en oeuvre sur minicalculateur,
- étude d'une structure multiprocesseur adaptée",
O. DOUCHIN, Thèse de Docteur 3ème cycle, Université de Nancy I, juin 1978.
- (5) "Conception et réalisation d'un moniteur temps réel pour un système multimicroprocesseur", M.L. LAGIER, Thèse de Docteur-Ingénieur, Université de Nancy I, décembre 1976.
- (6) "L'utilisation du multibus. Le multibus et ses signaux. Le SBC 86/12, un microordinateur 16 bits sur une seule carte", A. SABATIER, *Electronique et Applications Industrielles*, n° 264, 263, 270.
- (7) "La robotique industrielle à la Régie Renault", J. PASSEMARD, 11th C.I.R.P. International Seminar on Manufacturing Systems, Nancy, juin 1979.
- (8) "Système à multimicroprocesseurs pour un dialogue avec un robot", D. DOURS, R. FACCA, Laboratoire C.E.R.F.I.A., Toulouse, C.I.R.P. Nancy, juin 1979.

- (9) "Commande et régulation par calculateur numérique", C. FOULARD, S. GENTIL, J.P. SANDAZ, Editions EYROLLES.
- (10) "Images et ordinateurs", P. MORVAN, M. LUCAS, Larousse, Université.
- (11) "Eléments d'informatique", C. CORGE, Larousse, Université.
- (12) "Structure et fonctionnement des ordinateurs", J.P. MEINADIER, Larousse, Informatique.
- (13) "Machines informatiques", H. BOUCHER, Cours internes, E.N.S.T.A.

A.1.

ANNEXE TECHNIQUE

- OPREQ : OPERATION REQUEST. Le signal généré sur cette terminaison accompagne toute opération externe et il prend alors la valeur 1 logique durant 1,5 période d'horloge environ.
- $\overline{M}/\overline{IO}$: MEMORY/INPUT - OUTPUT. Indique le type d'entrée/sortie en cours.
M ou 1 logique $\rightarrow$ Mémoire,
 $\overline{IO}$ ou 0 logique $\rightarrow$ Périphérique.
- $\overline{W}/\overline{R}$: WRITE - READ. Précise le sens du transfert.
0 logique lecture,
1 logique écriture.
- $\overline{WRP}$: WRITE - PULSE. Impulsion de synchronisation des opérations d'écriture. Elle est générée après l'apparition du signal d'écriture, au bout d'un temps fixé par une température interne. Elle dure alors un temps d'horloge
- $\overline{E}/\overline{NE}$: EXTENDED - NON EXTENDED.
- $\overline{D}/\overline{C}$: DATA - CONTROL.

Ces informations sont disponibles sur les terminaisons ADR 13 et ADR 14 lors des entrées-sorties périphériques ($\overline{IO}$) et permettent de discriminer les deux types d'instructions d'entrée/sortie et la nature des octets véhiculés par le bus.

- RUN/WAIT : Cette terminaison fournit l'état du microprocesseur.
RUN 1 logique exécution en cours,
WAIT 0 logique état statique.
- INTACK : INTERRUPT - ACKNOWLEDGE.

Ce signal prend l'état 1 lorsqu'une demande d'interruption est acceptée.

A.2.3. - Signaux de services en entrée

Six entrées de commande sont utilisables.

- * Les deux premières agissent sur l'état des bus du microprocesseur.

- $\overline{ADREN}$: ADDRESS ENABLE.
- $\overline{DBUSEN}$: DATA BUS ENABLE.

La mise à l'état 1 de l'un de ces signaux produit la mise en l'état "haute impédance" du bus concerné.

- * Une autre intervient sur le séquençement des opérations avec l'environnement et permet un fonctionnement asynchrone.

- $\overline{OPACK}$: OPERATION ACKNOWLEDGE.

Réponse de l'extérieur à une demande d'OPREQ.

Si cette entrée est à l'état 1, le processeur, tout en restant dans l'état RUN, se suspend sur l'opération en cours et le signal OPREQ reste haut. Lorsque l'entrée prend l'état 0 le processeur exécute l'opération.

- * Les trois dernières entrées de commandes sont les suivantes :

- RESET : crée le retour en l'état initial, par la remise à zéro du compteur ordinal et l'autorisation des interruptions.
- $\overline{PAUSE}$: si ce signal est à l'état haut, le processeur est dans l'état RUN, et s'il prend l'état bas, le microprocesseur termine l'instruction en cours et se met dans l'état WAIT.

ANNEXE A

A - DESCRIPTION DU MICROPROCESSEUR 2650

A.1. - Principales caractéristiques du microprocesseur

- . Exécution en parallèle sur 8 bits,
- . Un registre d'usage général de 8 bits,
- . Deux bancs de 3 registres, usage général et indexe, de 8 bits,
- . Une pile interne autorisant 8 imbrications,
- . Un pointeur de pile sur 3 bits,
- . Un mot d'état (PSW) comprenant : retenue, débordement, signe d'un résultat, masque d'interruption, retenue intermédiaire,
- . Une ligne d'entrée et une ligne de sortie série,
- . Une interruption,
- . Un bus de données bi-directionnel 8 bits,
- . Un bus adresse sur 15 bits.

Le microprocesseur est réalisé en technologie MOS à canal N, tension d'alimentation 0, +5 V. Il nécessite une horloge monphasée générant des impulsions positives à une fréquence maximale de 1,25 MHz.

A.2. - Les terminaisons

A.2.1. - Les bus adresses et données

- ADR 0/ADR 14 : correspondent aux 15 bits du bus adresses. Ils permettent d'adresser 2¹⁵ = 32 768 octets (32 k octets) organisés en pages de 4 k octets.

Sur les lignes 0 à 7 transitent les adresses de périphériques dans le cas des entrées-sorties, les lignes 13 et 14 précisent alors le type de l'entrée-sortie.

- DBUS 0/DBUS 7 : 8 lignes d'entrée-sortie bidirectionnelles véhiculent les données et les instructions entre le microprocesseur et l'environnement.

A.2.2. - Signaux de service issus du processeur

8 terminaisons fournissent ces informations complémentaires soit pour préciser l'état du microprocesseur ou le type d'opération en cours, soit pour coordonner les opérations externes.

- INTREQ : INTERRUPT REQUEST.

La mise à zéro de cette entrée entraîne la mise en oeuvre de la procédure d'interruption si celles-ci ne sont pas masquées.

A.2.4. - Signaux particuliers

Les deux dernières terminaisons du microprocesseur sont respectivement en entrée et en sortie le reflet de deux bits du mot d'état du processeur. Leur utilisation principale correspond à des entrées-sorties séries.

- SENSE : Ligne d'entrée crée l'état du bit SENSE du PSW.

- FLAG : Ligne de sortie état du bit FLAG du PSW.

A.3. - Présentation logique

Le microprocesseur 2650 offre 7 registres adressables et utilisables par 4 (le registre R0 et l'un des deux bancs de 3 registres R1, R2, R3), et un mot d'état (PSW) réparti sur deux octets qui sont adressables séparément.

Ces informations sont les seules accessibles par programme. Ainsi l'on ne peut connaître, ni la valeur du compteur ordinal, ni le contenu des 8 mots de la pile des adresses de retour, utilisée lors des appels de sous-programme et des déroutements d'interruptions.

Le microprocesseur 2650 dispose de 74 instructions exécutables, ce qui représente 190 possibilités différentes, compte tenu des modes d'adressage et des modificateurs.

Nous allons présenter successivement le mot d'état du processeur, les modes d'adressages et les différents types d'instructions.

A.3.1. - Le mot d'état du processeur PSW (Programme Status Word)

Ce mot est présenté sous la forme de deux octets, adressables directement.

PSWL

7	6	5	4	3	2	1	0
S	F	II	-	-	SP2	SP1	SP0

PSWL

7	6	5	4	3	2	1	0
CC1	CC0	IDC	RS	WC	OVF	COM	C

Les informations contenues sont les suivantes :

- PSWL S - SENSE : reflet de l'entrée associée.

F - FLAG : commande de la sortie associée.

II - Inhibition des interruptions : toutes les interruptions sont masquées si ce bit est à 1.

SP - Pointeur de pile : valeur 0 à 7.

- PSWL CC - Code condition : positionné après les instructions arithmétiques, logiques et de chargement. Les valeurs possibles sont selon l'instruction :

. 00 - registre nul ou égalité,

. 01 - registre positif ou supérieur à la référence,

. 10 - registre négatif ou inférieur à la référence.

IDC - CARRY intermédiaire : pour l'arithmétique décimale.

RS : Sélection de l'un ou l'autre des bancs de registres.

WC : Selon sa valeur les opérations arithmétiques sont effectuées avec ou sans CARRY.

OVF - OVERFLOW : Débordement lors d'opérations arithmétiques ou logiques.

COM : Permet de préciser si l'on désire qu'une comparaison soit arithmétique ou logique.

C - CARRY : Retenue lors des opérations arithmétiques ou dernier bit sorti lors des décalages. Ce bit n'est utilisé dans les opérations en chaîne que si WC = 1.

A.3.2. - Les modes d'adressage

Les quatre modes d'adressage qui peuvent être utilisés, sont les suivants :

- mode registre,
- mode immédiat,
- mode relatif,
- mode absolu.

Toutes les instructions utilisant les modes relatif et absolu peuvent être spécifiées avec ou sans indirection. Une autre spécification autorise l'indexation pour les instructions sur mots mémoire.

A.3.3. - Les instructions

Les instructions occupent, suivant le mode d'adressage utilisé, de un à trois octets.

Les instructions arithmétiques se réduisent aux instructions d'addition et de soustraction. Toute les opérations logiques peuvent être exécutées ainsi que des décalages d'un bit.

Il est intéressant de noter le nombre important de branchements et les possibilités offertes : branchement selon la valeur d'un registre ou du code condition, selon la valeur d'un registre préalablement décrémenté, incrémenté ou indexé.

De même, il est possible lors des opérations sur mot mémoire, en mode absolu, de spécifier une auto-incrémentation ou décrémentatation du registre utilisé en index.

Les déplacements qui peuvent être spécifiés sont, selon le mode d'adressage :

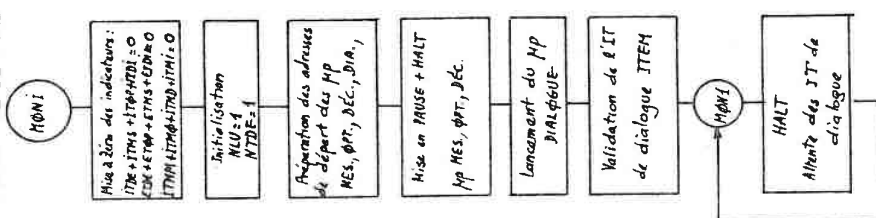
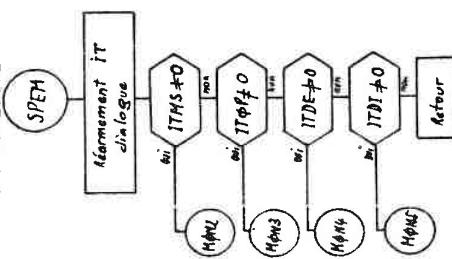
- mode relatif : -64, +63,
- mode absolu pour les instructions de branchement : 0, 32 767,
- mode absolu pour les instructions autre que branchement : 0, 4 095.

Pour ce dernier groupe d'instructions, la mémoire ne peut être adressée en entier que par une indirection.

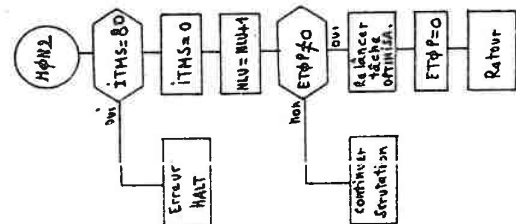
Lors des appels de sous-programmes, l'adresse courante est rangée automatiquement dans la pile puis restituée lors de l'exécution d'une instruction de retour de sous-programme.

ORGANIGRAMMES DES DIFFERENTES TACHES

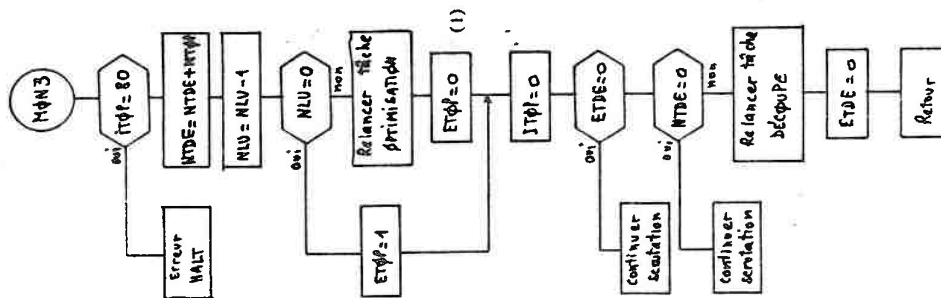
- MONITEUR A.5
- TÂCHE MESURE A.7
- TÂCHE OPTIMISATION A.9
- TÂCHE DÉCOUPE A.14

INITIALISATION ET
PROGRAMME DE FONDSOUS-PROGRAMME
IT. DIALOGUE

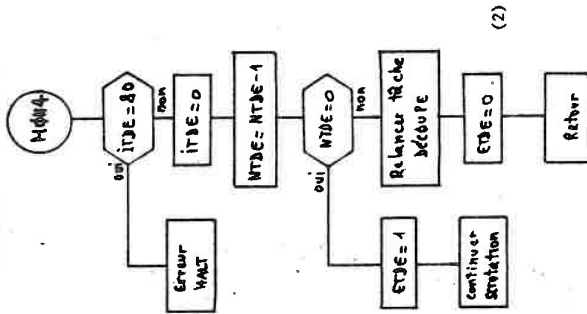
IT. DIALOGUE MESURE



IT DIALOGUE OPTIMISATION



IT DIALOGUE DECOUPE



- Les indicateurs ITMS, ITOP, ITDE, ITDI servent à coder les messages émis par une tâche à destination du Moniteur.

- Les indicateurs ITMI, ITMO, ITMD, ITMI servent à coder les messages émis par le Moniteur à destination des tâches.

(1) ETOP=0 il y a des planches à optimiser

ETOP=1 il n'y a plus de planches à optimiser.

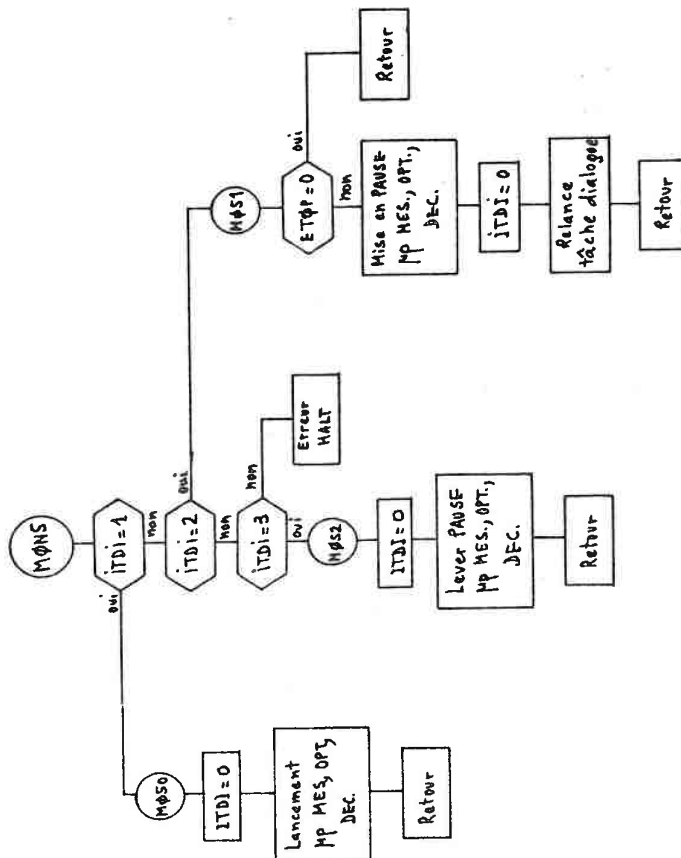
(2) ETDE=0 il reste des coupes à faire

ETDE=1 il n'y a plus de coupes à faire, la tâche est en Attente.

MONITEUR

INITIALISATION S PROG. DE FOND
DIALOGUE MESURE.OPTI.,DECOUPE

IT DIALOGUE DIALOGUE



-ITDI=1 Lancement du système après mise sous tension. La tâche DIALOGUE vient de vérifier qu'il existe au moins un tableau de découpe et que les paramètres soit mémorisés.

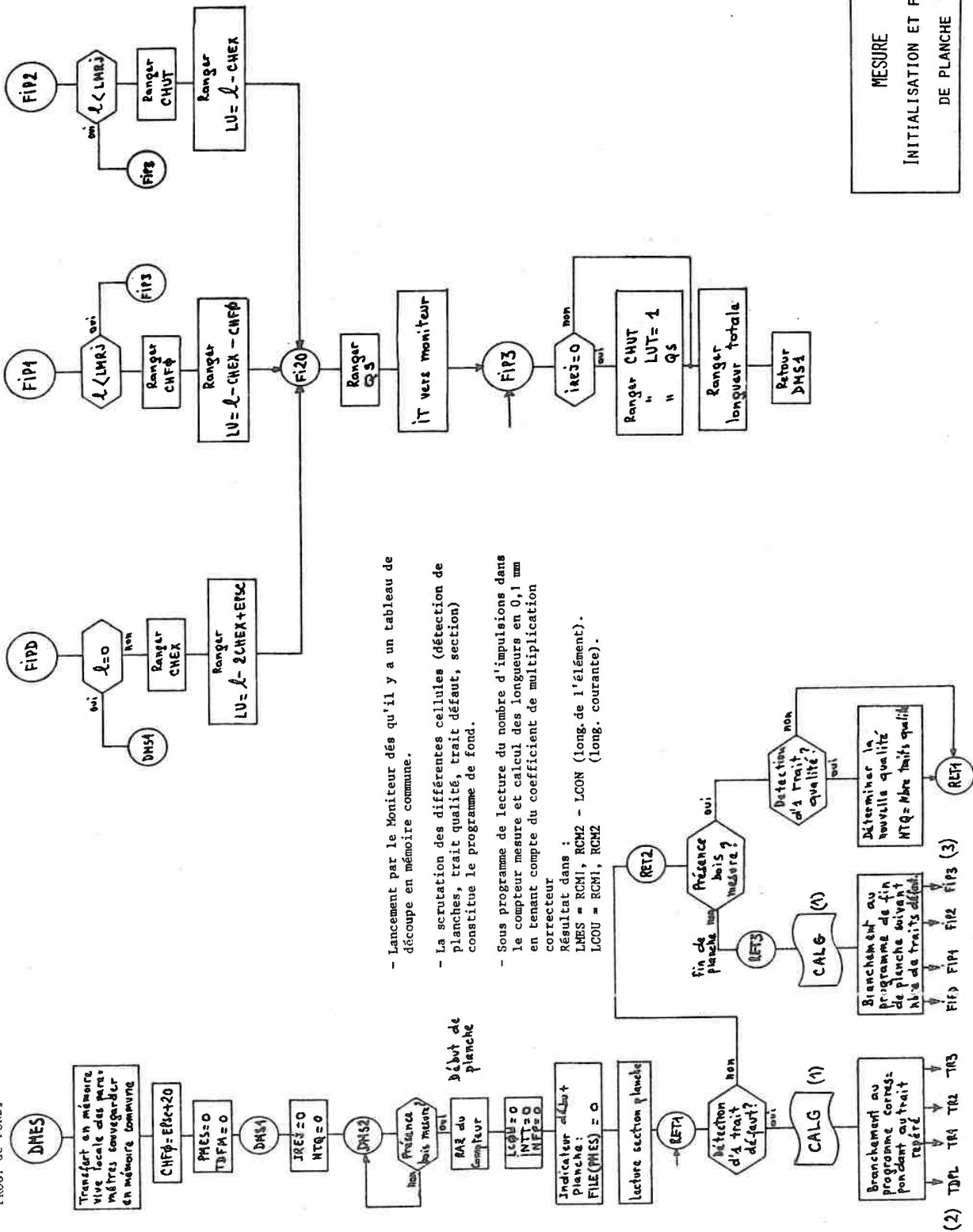
-ITDI=2 Demande de modification de tableaux en Mémoire Commune. Mise en pause des micros MESURE, OPTI, DECOUPE à condition qu'il n'y ait plus de longueurs utiles en attente d'optimisation.

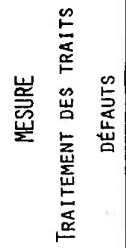
-ITDI=3 Fin de la modification et réactivation des micros MESURE, OPTI, DECOUPE.

MONITEUR
DIALOGUE DIALOGUE

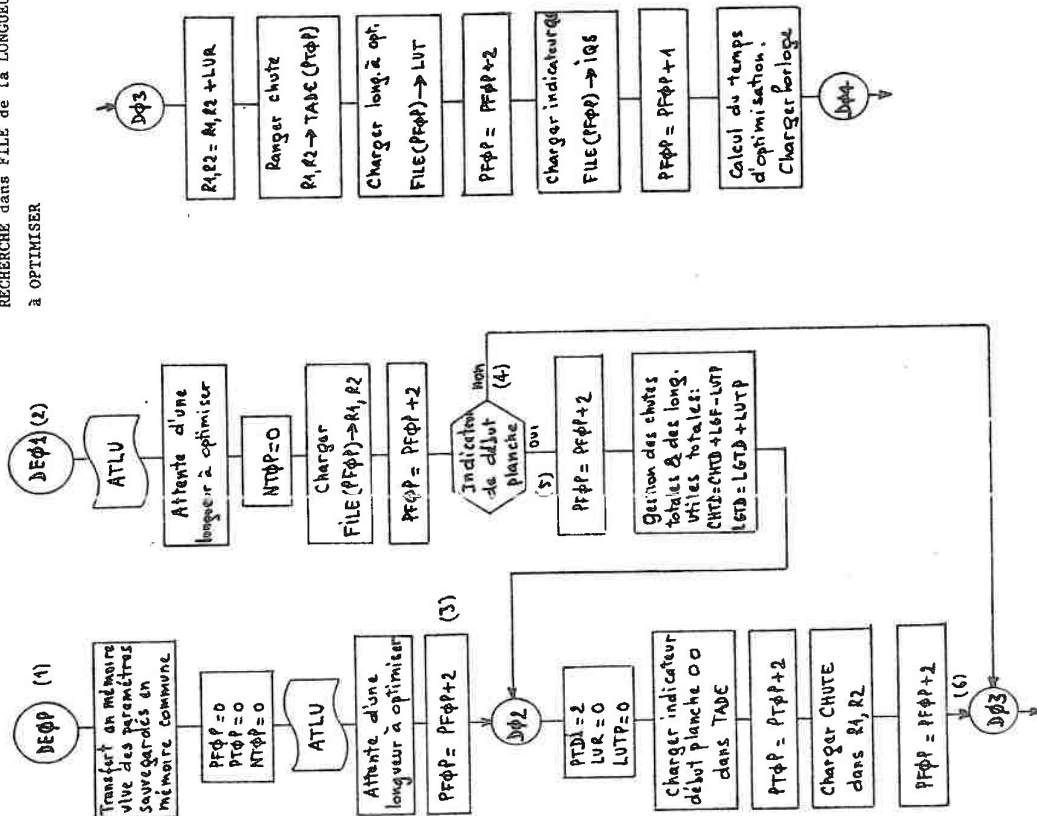
INITIALISATION et

PROC. de FOND.





RECHERCHE dans FILE de la LONGUEUR
à OPTIMISER



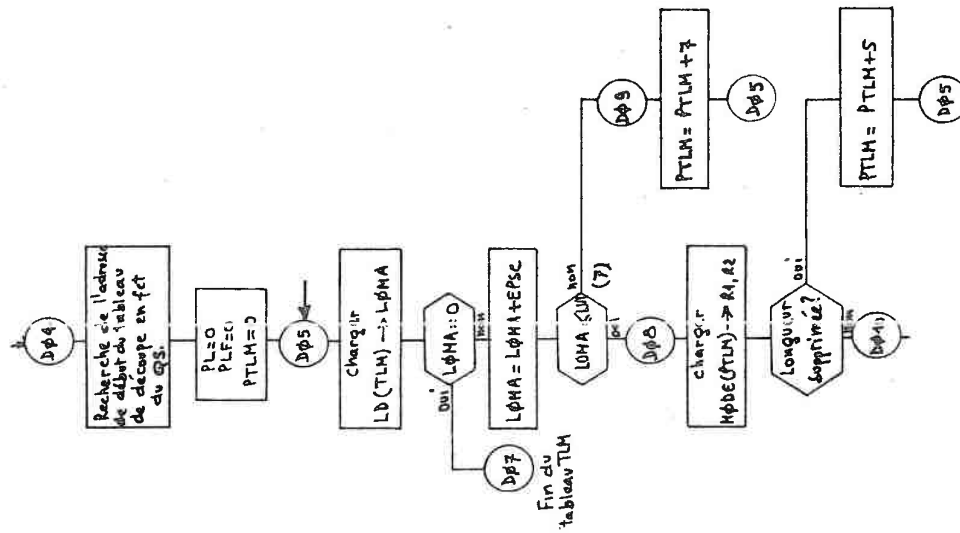
Une horloge programmable limite le temps d'optimisation en fonction de la longueur utile et de la vitesse de déplacement du tapis.

- (1) L'entrée DE0P, utilisée en début du programme d'optimisation, permet certaines initialisations.
- (2) L'entrée DE01 est ensuite utilisée pour l'optimisation courante.
- (3) Saut 00, début planche. PF0P pointe maintenant sur CHUTE.
- (4) PT0P pointe sur LU ; R1, R2 contiennent CHUTE.
- (5) R1, R2 contiennent long. totale LCF ; PF0P pointe maintenant sur CHUTE.
- (6) PT0P P pointe maintenant sur LU.

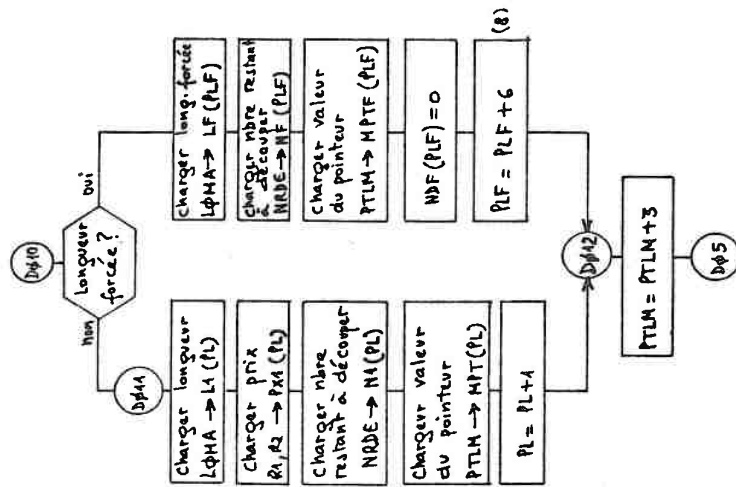
OPTIMISATION

RECHERCHE DANS FILE

TRANSFERT du TABLEAU TLM dans les TABLEUX L et LF



(7) Longueur marchande trop grande.



(8) PLF pointe maintenant sur le prochain LF

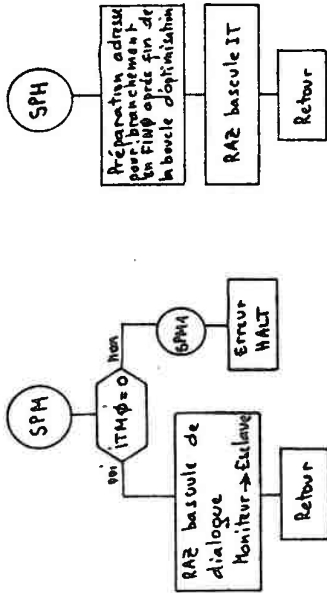
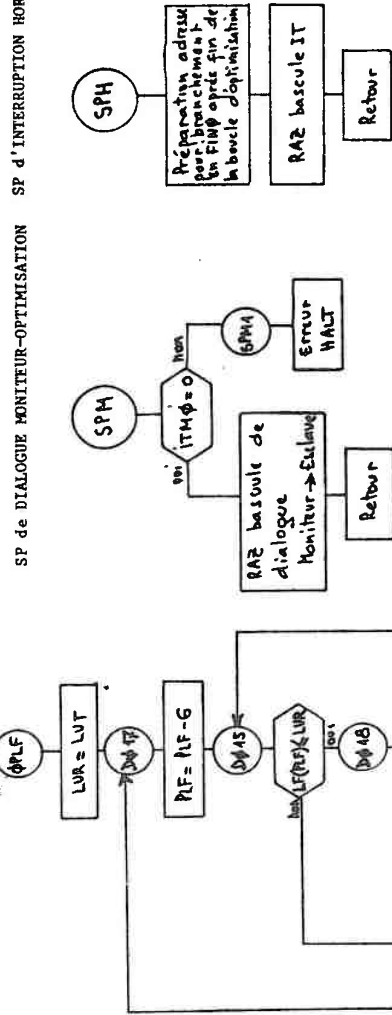
OPTIMISATION
CHARGEMENT TABLEUX L ET LF

OPTIMISATION de la LONGUEUR UTILE ET RANGEMENT
du RESULTAT dans TADE

SP d'OPTIMISATION des LONGUEURS FORCEES

SP de DIALOGUE MONITEUR-OPTIMISATION

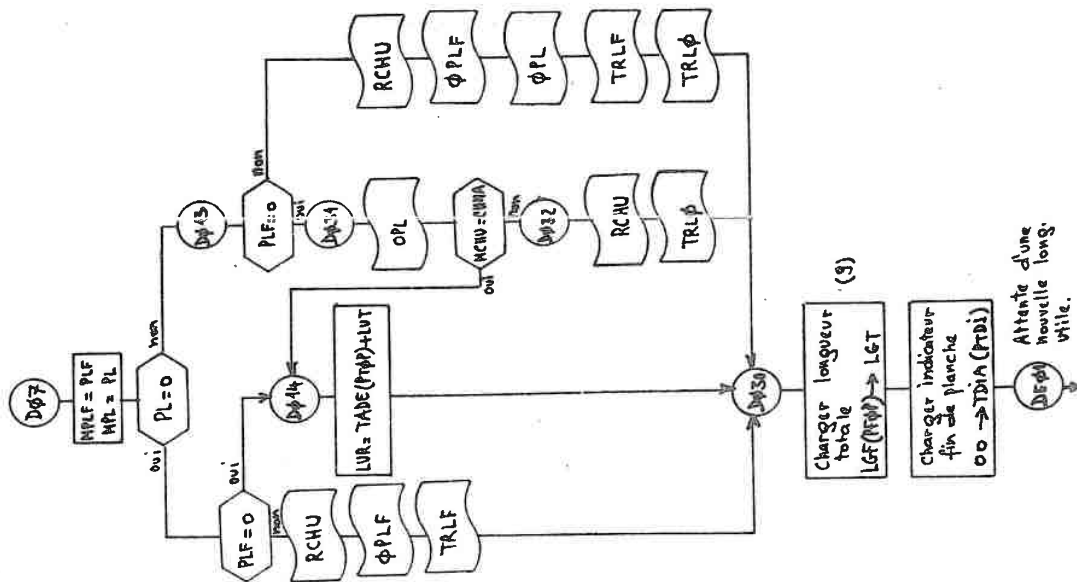
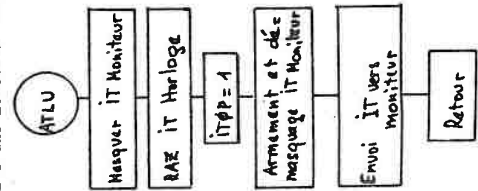
SP d'INTERRUPTION HORLOGE



SP de RANGEMENT de la CHUTE dans TDIA



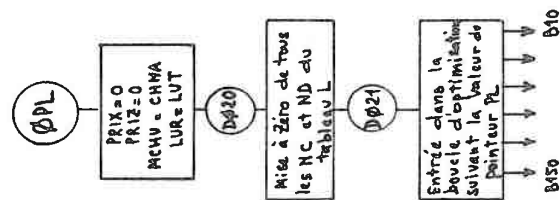
SP ATTENTE d'une LONGUEUR UTILE



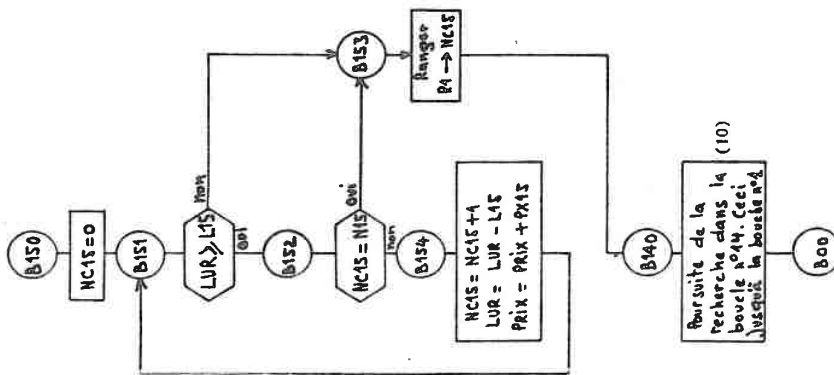
(9) Systématiquement fait pour chaque longueur utile.
La dernière longueur totale chargée en fin de
planche est la seule valable.

OPTIMISATION
OPTIMISATION ET RANGEMENT

INITIALISATION de la BOUCLE

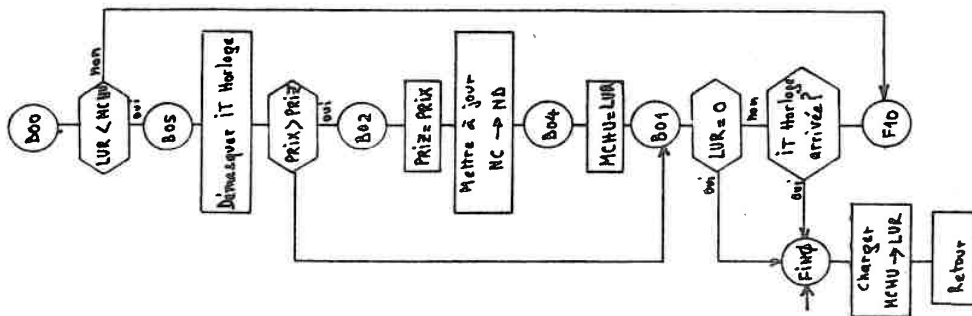


BOUCLE d'OPTIMISATION

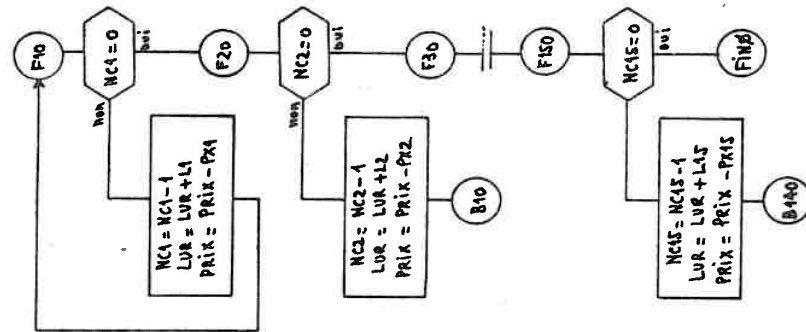


(10) La boucle n° 1 (Entrée B10) traite la plus petite longueur à optimiser ; la boucle n° 15 (Entrée B150) traite la plus grande.

ANALYSE du RESULTAT APRES PASSAGE dans la BOUCLE

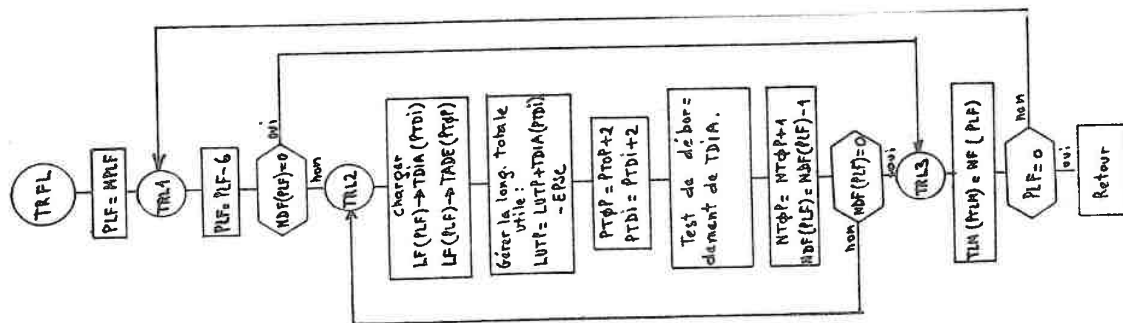


DECREMENTATION de la PLUS PETITE LONGUEUR TROUVEE et RETOUR dans la BOUCLE d'OPTIMISATION



OPTIMISATION
SP D'OPTIMISATION

RANGEMENT des LONGUEURS FORCÉES OPTIMISÉES
dans les TABLEAUX TADE et TDIA et MISE à
JOUR du NOMBRE RESTANT à DECOUPER NRDE (TLM).



RANGEMENT des LONGUEURS NON FORCÉES OPTIMISÉES
dans les TABLEAUX TADE et TDIA et MISE à JOUR
du NOMBRE RESTANT à DECOUPER NRDE (TLM).

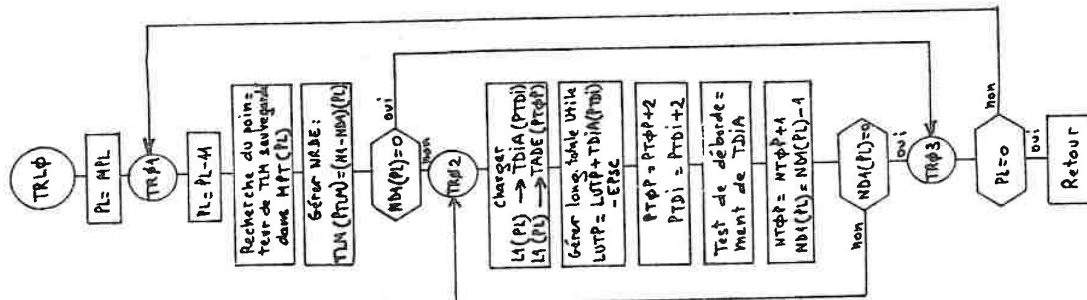


Tableau L (utilisé par la boucle d'optimisation des longueurs non forcées).

L1	longueur à découper (plus petite)
N1	nombre restant à découper
PX1	prix
MPT	mémorisation du pointeur dans TLM
NC1	nombre courant de longueurs
ND1	nombre de longueurs retenues
GTCA (B10)	adresse de début de boucle

L2
N2
—
etc
—

L15 longueur à découper (plus grande)

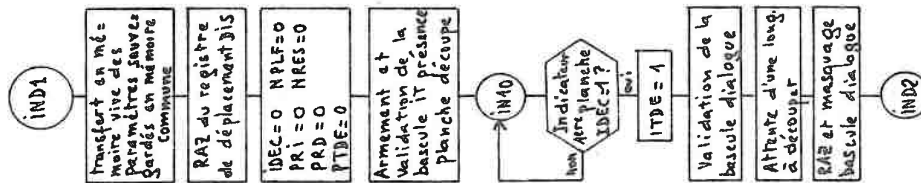
N15
PX15
MPT
NC15
ND15
GTCA (B150)

Tableau LF (longueurs forcées)

LF	longueur forcée
NF	nombre restant à découper
MPTF	mémorisation du pointeur dans TLM
NDP	nombre de longueurs à découper

OPTIMISATION
SP DE RANGEMENT DANS TADE

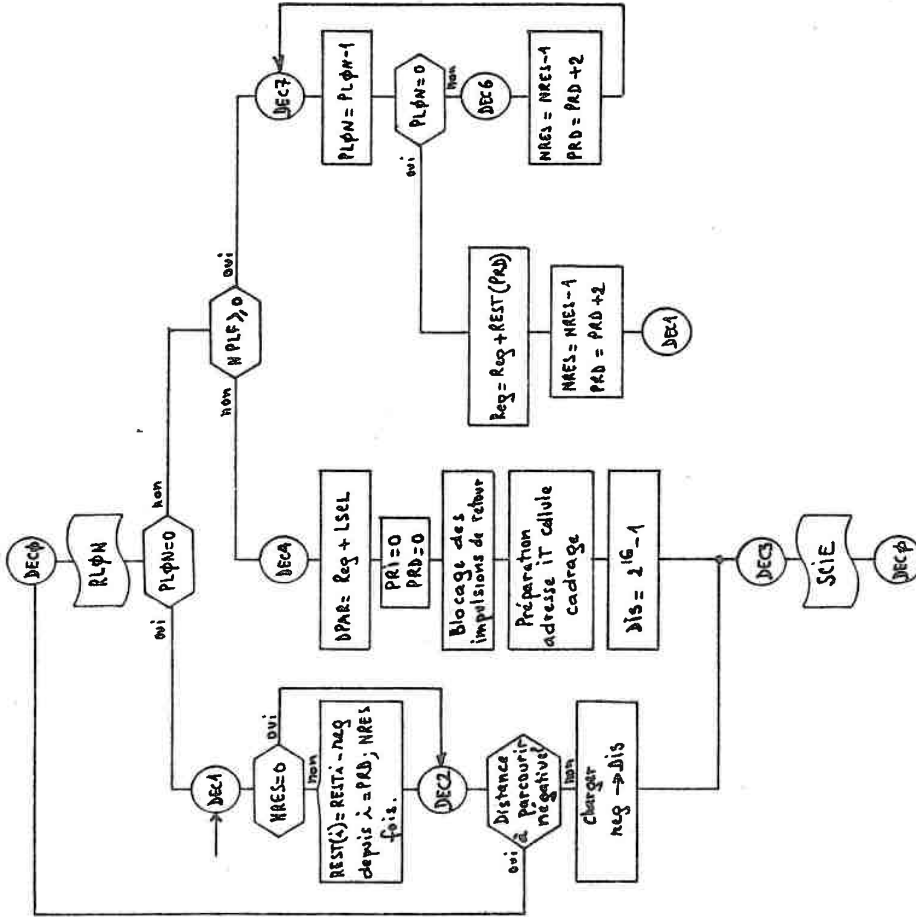
INITIALISATION et ATTENTE
d'une LONGUEUR à DECOUPER



(1) Recherche d'une longueur à découper dans le tableau TADE.

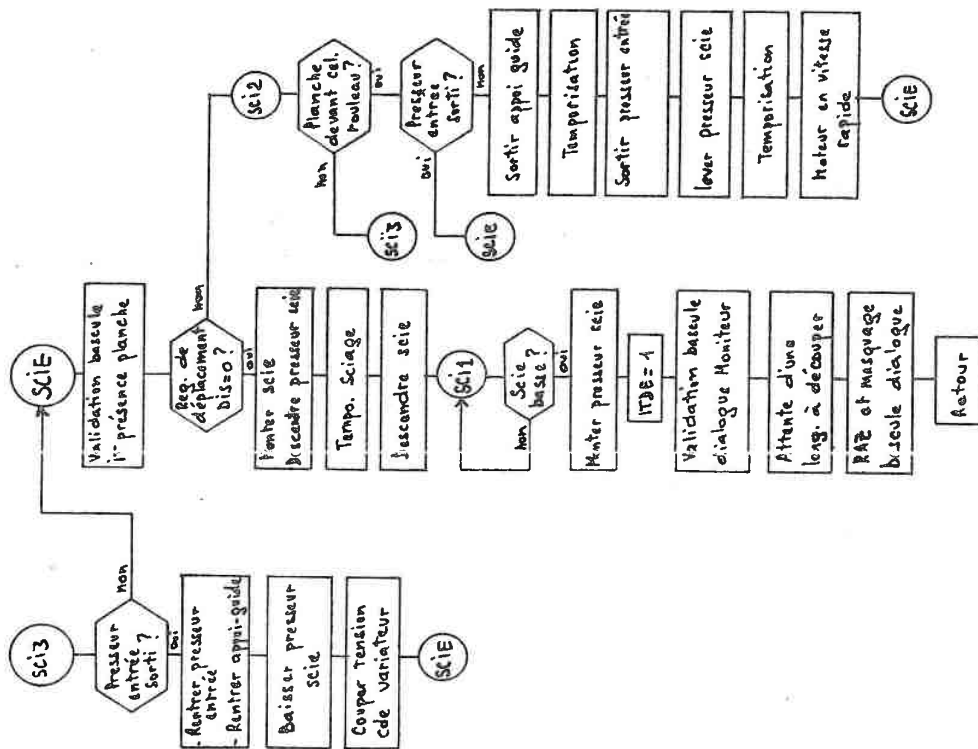
(2) Positionnement de la planche de cadrage.

DECOUPE COURANTE

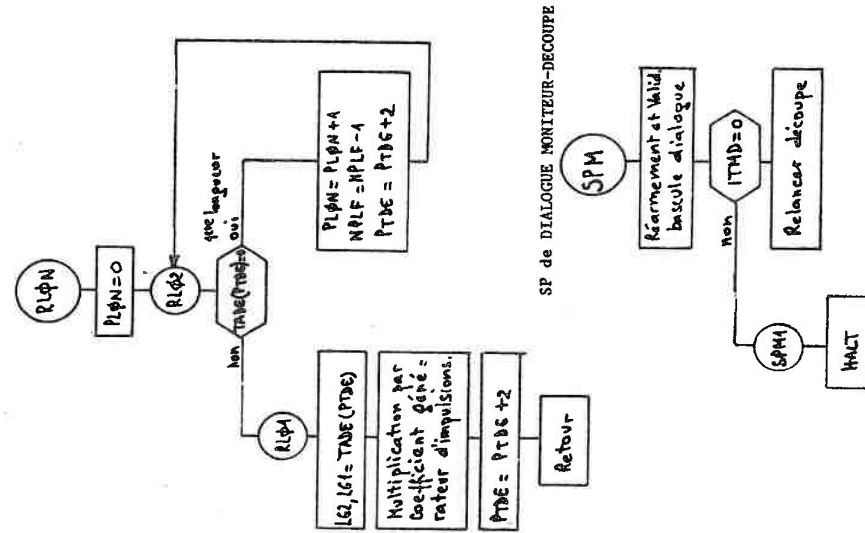


DECOUPE
INITIALISATION ET
DECOUPE COURANTE

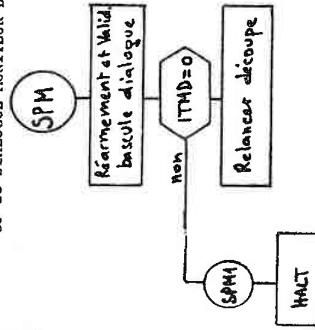
SCRUTATION du REGISTRE de DEPLACEMENT



RECHERCHE d'une LONGUEUR à DECOUPER
dans le TABLEAU TADE

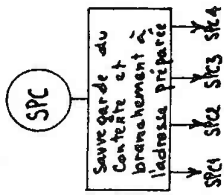


SP de DIALOGUE MONITEUR-DECOUPE

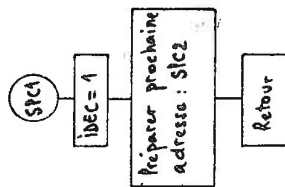


DECOUPE
SP. SCIE, RLON ET SPM

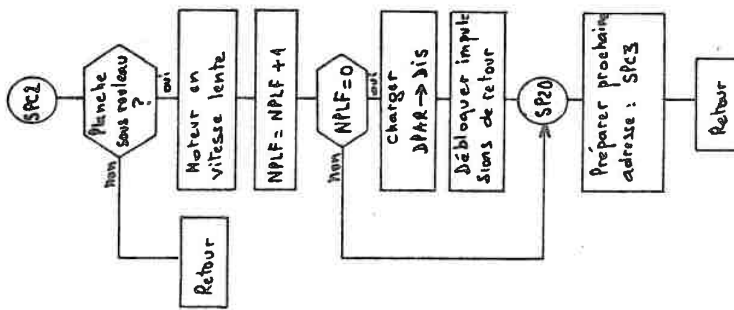
SP INTERRUPTION CELLULE DE CADRAGE



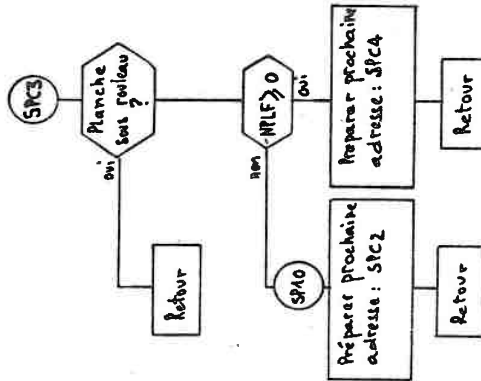
ARRIVEE PLANCHE de CADRAGE



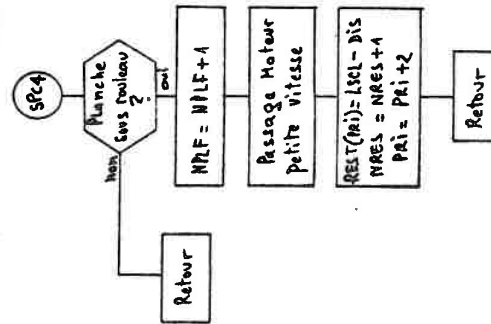
FIN de la PLANCHE de CADRAGE



ARRIVEE d'une PLANCHE COURANTE



FIN d'une PLANCHE COURANTE



DECOUPE
IT CELLULE DE CADRAGE

NOM DE L'ETUDIANT : DRAPIER BECHE Jean Pierre

NATURE DE LA THESE : DOCTEUR INGENIEUR en AUTOMATIQUE

VU, APPROUVE

et PERMIS D'IMPRIMER

NANCY LE 10 JAN 1980 06879

LE PRESIDENT DE L'UNIVERSITE DE NANCY I

