

# COLLOQUE NATIONAL SUR LE TRAITEMENT DU SIGNAL ET SES APPLICATIONS

NICE du 16 au 21 JUIN 75



INFORMATIQUE ET TRAITEMENT DU SIGNAL

C. BOZZO

GESTA-CAPCA - DCAN de TOULON

## RESUME

Les Systèmes Informatiques modernes associés à des processus physiques réels dans le cadre de problèmes de détection, d'estimation et de filtrage ou de commande ont des fonctions :

- de traitement de l'information (traitement de données et gestion de données),
- de traitement du signal en remontant vers l'amont et vers les senseurs.

Les fonctions de traitement du signal étaient de manière traditionnelle effectuées dans une technologie "câblée" en Analogique ou digital. Depuis 1970 le domaine d'application de l'informatique s'est étendu du traitement de l'information à une part plus ou moins importante du traitement du signal et ce sous le triple aspect :

- des fonctions d'acquisition,
- des fonctions de traitement,
- des fonctions de visualisation.

L'exposé présenté se propose d'effectuer une synthèse qui permette d'envisager :

- a) Les causes de l'introduction de l'informatique dans le traitement du signal : Ces causes sont de nature économique (coût des matériels informatiques) ou techniques.
- b) Les fonctions de l'informatique en traitement du signal, la nature des traitements effectués et leurs relations avec les fonctions de traitement de l'information.
- c) Les matériels dont dispose ou peut disposer le concepteur de système et les logiciels associés : microprocesseurs, miniordinateurs etc ... ainsi que la structure et l'organisation des processeurs bien adaptés au traitement du signal et l'architecture des systèmes informatiques correspondant.
- d) Les problèmes de disponibilité, de fiabilité et de reconfiguration.

## SUMMARY

The combination of new mathematical tools - the Fast Fourier Transform and the Theory of Sampled data Systems and of discrete optimal filtering - and specially designed computer hardware speeds up digital signal processing. The application of signal processors under microprogram control to acoustic and radar processing are now very important. This paper surveys basic digital signal processing algorithms. It proposes an analysis of signal processing architecture including parallel and "pipe-line" architectures and architectures consisting of a microprogrammed control processors and highly efficient signal processing arithmetic units which can be either an associative processor, parallel ensemble, matrix array processor or a vector processor.

Hardware technology surveys, firmware and microprogramming support, software discussions and problems of disponibility and reliability are also presented.

1. - INTRODUCTION DES TECHNIQUES NUMERIQUES ET DE L'INFORMATIQUE EN TRAITEMENT DU SIGNAL

1.1 - Classification de la nature des traitements effectués dans un système informatique en temps réel embarqué complexe

En allant des senseurs (Radar, Sonar etc ...) vers l'exploitation synthétique par un opérateur des informations traitées, la synthèse un système "temps réel" complexe conduit à la juxtaposition de trois classes d'activités :

- activités de traitement de signal,
- activités de traitement de l'information qui se décomposent en :
  - activités de traitement de données,
  - activités de gestion de données.

La nature des tâches correspondant à chacune de ces activités et les caractéristiques des ensembles de traitement de l'information associés, sont essentiellement différentes.

Le traitement du signal est constitué par "un ensemble de théories et de techniques qui visent essentiellement à extraire des renseignements utilisables d'une information altérée, soit par la présence de bruits, soit par l'effet d'une propagation aléatoire, soit encore par un mode de génération comportant également des éléments aléatoires" [3]. Les traitements à effectuer (cf. paragraphe 2) sont essentiellement des traitements arithmétiques et logiques et ils présentent un caractère répétitif marqué.

Les activités de traitement de l'information se situent en aval de l'activité de traitement du signal. Il s'agit en effet de trier, d'organiser, de présenter des informations qui ont été détectées, estimées, filtrées, dans le cadre des activités de traitement du signal et qui présentent donc un certain caractère de fiabilité. Ces mêmes informations peuvent d'ailleurs être à l'origine de commandes, soit par l'intermédiaire d'un opérateur, soit dans le cadre d'une chaîne directe. Il est possible de constater que l'on retrouve les deux fonctions essentielles d'un processeur digital moderne : traitements arithmétiques et traitements logiques.

Les traitements logiques correspondent aux activités de gestion de données (trier, organiser, présenter ...). Les moyens informatiques associés doivent permettre les opérations sur listes, sur chaînes, sur variables logiques et la gestion de fichiers importants, etc ... Une application classique des activités de gestion de données est la tenue d'une situation tactique, ou d'une situation de contrôle aérien etc ...

Les traitements arithmétiques correspondent aux activités de traitement de données (commande en temps discret, traitement de cohérence au sens général etc ...) Les moyens informatiques associés doivent permettre les opérations arithmétiques simples (addition, multiplication) ou complexes, les périphériques temps réel, les activités d'acquisition ou de commande etc ... Des applications classiques des activités de traitement de données sont l'extraction en Radar ou Sonar (cohérence plot), la commande de système échantillonnés etc ...

Si l'informatique s'est introduite de manière naturelle depuis une dizaine d'années dans les fonctions de traitement de l'information, il n'en est pas de même en traitement du signal, domaine dans lequel les moyens de traitements étaient jusqu'à ces dernières années essentiellement analogiques et figés dans leur structure câblée.

L'introduction des techniques numériques et de l'informatique résulte de la conjonction de deux évolutions qui ne sont pas entièrement indépendantes :

- évolution des concepts fondamentaux,
- évolution des techniques de traitement numérique de l'information sur le plan technologique et économique.

1.2 - Influence de l'évolution des concepts fondamentaux

Il n'est pas possible de traiter ici de l'ensemble des opérations de traitement que l'on peut faire subir à un signal. Ce qui va suivre risque donc de faire frémir les spécialistes.

Les domaines d'élection du traitement du signal [3] sont complémentaires. Il s'agit de :

- la détection de la présence d'un signal noyé dans un bruit avec le minimum de fausses alarmes, pour un maximum de probabilités de détection,
- la caractérisation et l'estimation des paramètres d'un signal (doppler, distance etc ...).

Le traitement du signal fait donc appel au calcul des probabilités et plus particulièrement à la théorie des fonctions aléatoires et à la théorie statistique de la détection. L'approche très souvent adoptée consiste à exploiter la dualité entre les domaines de description temporel et fréquentiel d'un signal, dualité qui s'exprime par la notion de transformation de Fourier. On peut donc raisonner indifféremment dans l'un ou l'autre domaine.

Une opération fondamentale qui intervient dans le domaine temporel est la corrélation qui permet d'effectuer une estimation statistique de la "ressemblance" entre deux fonctions aléatoires (intercorrélacion) ou de la permanence au cours du temps d'une même fonction (autocorrélacion). On peut d'ailleurs montrer que l'intercorrélacion :

$$\varphi_{xy}(\tau) = \lim_{T \rightarrow \infty} \frac{1}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} x(t) y(t+\tau) dt$$

correspond à la convolution de x(-t) avec y(t) ou au produit de Y(f) transformée de Fourier de y(t) par le complexe conjugué de X(f).

Pour détecter un signal non aléatoire dans un bruit à densité spectrale uniforme, le filtre optimal (filtre adapté) a pour gain complexe la transformée de Fourier conjuguée du signal à un facteur réel près. Dans le domaine temporel l'opération correspondante consiste à effectuer une convolution de l'information à traiter avec la copie du signal.

Pour détecter un bruit de densité spectrale donnée dans un bruit parasite, il existe également un filtre optimal qui est fonction des deux spectres, l'utile et le parasite. L'opérateur de détection doit être adapté en fonction de la connaissance des densités spectrales et en particulier de la détermination en temps réel d'une référence bruit seul qui fait intervenir des opérations d'estimation des caractéristiques statistiques du bruit parasite. Ces opérations peuvent être effectuées en élaborant, par exemple, les moments (moyenne, variance etc ...) sur des tranches de l'information à traiter ou en appliquant les théories de filtrage et d'estimation modernes, qui conduisent en fait à effectuer une convolution de l'information à traiter par la réponse impulsionnelle du filtre optimal déterminé.

L'évolution importante des concepts fondamentaux a consisté, à partir des années 1955-1960, à



transposer dans le domaine discret les principales notions (corrélation, convolution, filtrage optimale, Transformation de Fourier etc ...) qui étaient appliquées dans le domaine continu. Cette évolution est, bien entendu, liée à l'apparition de machines de traitement numériques, mais il ne semble pas exact de dire qu'elle en résulte complètement.

Si l'on considère les relations fondamentales de la Transformation de Fourier, de la convolution et de la corrélation, on constate que les opérations élémentaires à effectuer sont :

- le retard par une quantité continue ou discrète,
- la mémorisation de coefficients réels ou complexes,
- la multiplication,
- l'intégration ou la sommation.

Dans les techniques digitales les retards et les mémorisations, ainsi que les opérations arithmétiques sont facilement obtenues, les opérations arithmétiques ne posent que le problème de leur vitesse d'exécution. Cet avantage fondamental doit d'ailleurs être associé aux possibilités d'exécution d'opérations logiques d'un processeur qui permettent d'effectuer les traitements décisionnels que l'on rencontre dans certains problèmes de détection, d'estimation, d'extraction traitements qui sont pratiquement impossibles à réaliser par la voie analogique.

Mais il faut également noter que sous l'impulsion d'un certain nombre de chercheurs JURY, RAGAZZINI, TOUTY, TZYPKIN, VOLGUINE, ZADEH ... pour les systèmes déterministes KALMAN, BUCY, MEHRA, KAILATH, HO, SAGE ... pour les systèmes stochastiques, GOLD, RADER, COOLEY, TUKEY, PAPOULIS, STEIGLITZ ... pour le traitement du signal et les méthodes associées, les concepts théoriques en matière de processus en temps discret, de traitement et de filtres numériques, ont très rapidement évolué de la transposition directe des méthodes de traitement en temps continu au développement de méthodes originales et dont l'apport théorique est important (théorie des systèmes échantillonnés, estimation et filtrage optimale en temps discret, Transformée de Fourier Discrète etc ...) et repose sur le fait que certains concepts fondamentaux sont beaucoup plus aisés à exprimer dans le domaine discret (sur le plan théorique comme sur le plan pratique), que la connaissance en temps continu d'un signal n'est pas strictement utile sur le plan théorique (théorème de Shannon) etc ...

### 1.3 - Influence de la technologie facteurs d'évolution technologique et économique

Les raisons technologiques de l'introduction des techniques digitales et de l'informatique dans le traitement du signal sont bien connues. Plusieurs facteurs ont apporté d'importants bouleversements dans le domaine des processeurs numériques depuis une dizaine d'années.

#### 1.31 - Evolution de la technologie

Le point le plus important concerne surtout l'intégration de plus en plus poussée et l'amélioration des performances des composants. Cette évolution a conduit récemment à la réalisation de boîtiers LSI (large scale integration) contenant une unité arithmétique et logique et un ensemble de registres. Ces boîtiers appelés microprocesseurs sont à l'origine d'une nouvelle génération de systèmes, les micro-ordinateurs. Parallèlement, se sont développées d'autres techniques de conceptions de processeurs et en particulier la microprogrammation : chaque instruction de l'ordinateur est décomposée en une série d'opérations élémentaires dont l'ensemble constitue le jeu de micro instructions de la

machine. Une mémoire morte de l'ordinateur définit la série de micro-instructions devant être exécutées pour chaque instruction : c'est la mémoire de microprogrammation. La microprogrammation est donc une technique "qui permet de substituer au jeu de micro-instructions de la machine, dépendant de sa structure, un jeu d'instructions généralement plus évolué qui a un caractère un peu artificiel et tout à fait évolutif" [11]. Un microprocesseur peut être microprogrammé, mais cela n'est pas toujours le cas.

Le microprocesseur désigne donc un ou plusieurs boîtiers à 40 broches contenant chacun un circuit intégré à grande échelle qui regroupe à lui seul toutes les fonctions d'un processeur classique ("CPU chips" central process unit). Les circuits LSI actuels font déjà tenir plus de 10.000 transistors sur une puce de 5x5 mm. L'évolution récente de la technologie MOS (Metal Oxide Semi conductor) et de l'une de ses variantes SOS (Silicon On Sapphire) ont conduit à ce type de circuits. Malheureusement le microprocesseur tel qu'il existe actuellement, ne peut prétendre remplacer le processeur à hautes performances qui équipe la plupart des ordinateurs et mini-ordinateurs. Sa haute intégration, si elle réduit les encombrements, a limité la vitesse d'exécution des instructions et ses possibilités en matière d'E/S. La technologie MOS/LSI est loin en effet d'égaliser les circuits TTL (Transistor Transistor Logic) sur le plan de la vitesse de commutation. Il faut cependant noter que l'utilisation par les constructeurs de la technologie N.MOS (MOS Canal N) permettra d'augmenter la vitesse.

Les ordinateurs classiques ont été bâtis autour de mémoire à tores. Dans ce type de mémoire les organes de sélection représentent un coût qui n'est pas proportionnel à la taille mémoire et qui est d'autant plus élevé que celle-ci est petite. Par contre, dans les mémoires à semi-conducteurs les organes d'accès sont en grande partie inclus dans les circuits de mémoire. Le coût de celle-ci est donc proportionnel à sa taille. (RAM Random Access Memory). Les mémoires vives à semi-conducteurs sont plus rapides que les mémoires à tore (100 à 300 ns) mais très volatiles : l'utilisation de mémoires mortes ne nécessite pas de disposer de périphérie de rechargement. Ces mémoires mortes peuvent être ordinaires (ROM Read Only Memory, PLA Programmable Logic Array) programmables (programmable Read Only Memory) ou reprogrammables (REPROM).

L'utilisation de mémoires mortes confère au système un aspect analogue à celui d'une logique câblée.

La figure ci-dessous [12] permet de caractériser l'évolution des performances de la logique et des mémoires au cours des dix dernières années.

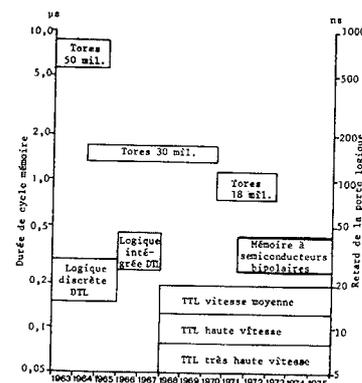


Figure 1  
Evolution des performances de la logique et des mémoires

1.32 - Evolution des coûts

La figure 2 extraite de [12] concernant le prix des composants par grandes quantités montre qu'en quelques années les coûts de la logique ont été divisés par 30 et ceux des mémoires par 6. Par contre, les prix des périphériques électromécaniques classiques ont très peu évolué depuis dix ans (dérouleurs, imprimantes, disques). Il faut noter que ces périphériques sont assez peu utilisés dans les applications de traitement de signal.

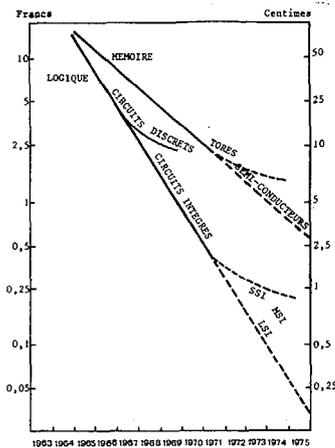


Figure 2

Evolution des coûts de la logique et des mémoires

1.33 - Les mini-ordinateurs

Pendant que l'informatique évoluait vers sa forme lourde et concentrée, un marché parallèle était en train de s'épanouir avec un dynamisme très important : celui des mini-ordinateurs. Les minis dont le développement technologique n'a pas été freiné par les problèmes économiques liés à la location des ordinateurs classiques ont fait de très grands progrès en matière de rapidité et de taille mémoire. Certains constructeurs [17] annoncent par exemple à leur catalogue (Interdata) des machines à mots de 32 bits, cycle de 800 nanosecondes et mémoires centrales jusqu'à un méga-octet. Ces machines sont d'autre part produites et vendues à des prix de plus en plus faibles : le PDP 8 de DEC a vu son prix de vente passer de 18.000 \$ en 1965 à 4.000 \$ en 1972.

Les mini-ordinateurs sont très bien adaptés aux problèmes de traitement de signal. Les causes principales de cette adaptation sont les suivantes :

- Aspects économiques - Depuis l'apparition des mini-calculateurs (1963-1964) le prix des processeurs "standards" diminue d'environ 20% par an. Ce prix est désormais assez faible pour qu'il soit possible d'envisager des structures multiprocesseurs complexes (cf. paragraphes 2 et 3) et pour que leur utilisation soit concurrentielle vis-à-vis du matériel câblé.

- Performances - Elles ont beaucoup augmenté depuis quelques années et, compte tenu de certaines organisations (cf. paragraphe 3), les performances des UAL et des mémoires sont très largement homogènes avec celles des plus grosses machines. Par contre, les performances des périphériques n'ont que peu évolué, les progrès dans le domaine de l'électromécanique étant plus lents. Cela n'est pas extrêmement gênant dans la mesure où les applications de traitement de signal ne font que très peu appel aux périphériques informatiques classiques.

- Encombrement - L'intégration très poussée et l'absence de périphériques électro-mécanique en traitement

du signal conduit souvent à des volumes unitaires très faibles pour les ensembles de traitement. Il est donc possible d'utiliser un grand nombre de processeurs simultanément.

- Aspects technologiques - L'évolution importante du niveau d'intégration conduit à une diminution du nombre de boîtiers de circuits logiques (le facteur peut être de l'ordre de 10) et du nombre de cartes par machine (quelques unités, mais de grandes cartes).

Une prédominance très nette de la technologie TTL peut être constatée (grande diffusion, prix faibles, game très riche etc ...). Les technologies MOS plus lentes et ECL plus rapides restent réservées à des applications particulières.

L'emploi de mémoires mortes (en technique MOS ou bipolaire ou mixte) permet l'utilisation de plus en plus fréquente de la microprogrammation et donc de réalisation bien adaptées aux problèmes spécifiques de traitement du signal (cf. paragraphe 3.5). Les mémoires vives peuvent également être réalisées dans des technologies à semi-conducteurs.

Le logiciel est souvent très complet et possède une grande puissance pour les applications spécifiques pour lesquelles des langages et des moniteurs spécialisés ont été développés.

- Variété du choix - Les constructeurs sont très nombreux et les modèles très variés. Certains ont été produits à plusieurs milliers d'exemplaires dans des versions plus ou moins industrialisées ou militarisées qui permettent de développer des prototypes à moindre coût.

2. - ANALYSE DES TRAITEMENTS ET DE LEUR ORGANISATION

La structure et l'organisation actuelle des machines informatiques classiques conduisent à effectuer des traitements à caractère essentiellement séquentiel avec "des schémas de réalisation fortement multiplexés" [4]. L'ensemble de traitement est fortement centralisé et le multiplexage conduit très souvent à une impossibilité au niveau de l'exécution "simultanée" de nombreuses tâches élémentaires simples. Le problème fondamental est donc d'analyser les traitements, indépendamment de toute organisation du système de traitement associé et de détecter les parallélismes existants, les simultanités apparentes ou effectives etc ...

2.1 - Notion de parallélisme des traitements

Il faut tout d'abord noter qu'une confusion est très souvent effectuée entre :

- traitements "séquentiels",
- traitements "simultanés" : les tâches correspondantes doivent être prises en compte au même instant,
- traitements "concurrents" : les tâches correspondantes doivent être prises en compte pendant le même intervalle de temps,
- traitements "pipe-line" : (ou cascade) la même tâche est effectuée de manière répétitive sur un seul flot de données pendant un intervalle de temps donné.

L'approche la plus naturelle consiste à analyser les traitements de manière à détecter les parallélismes éventuels : une organisation hardware elle-même parallèle devrait alors permettre de résoudre les conflits qui se posent dans l'approche séquentielle et multiplexée classique. Malheureusement il n'est pas certain que cette idée "très naturelle" conduise au meilleur "profit marginal", si l'on tient compte de la complexité accrue des processeurs à organisation parallèle.



Cette analyse du parallélisme des traitements a été envisagée par de nombreux chercheurs et a conduit à certains travaux dont les résultats ne sont pas très convaincants [6] etc ... Comme le fait remarquer très justement ARTAUD [4] : "On peut penser que les raisons de ce manque relatif de succès sont dues à la nature des programmes qui ont été pris en compte, qui correspondent bien souvent à des algorithmes fondamentalement séquentiels".

En traitement du signal par contre les tâches simultanées (conduisant à des traitements par essences parallèles) existent très souvent, de par la nature même des senseurs considérés. Ces tâches appartiennent à trois catégories principales.

### 2.11 - Traitements "vectoriels ou matriciels" avec simultanéité

Il s'agit de traitements arithmétiques ou logiques simples effectués de manière simultanée en temps réel sur de nombreuses chaînes ou sur des flots de données indépendants. Les traitements sur chaque chaîne sont en général identiques. Les exemples de ce type de parallélisme sont très courants en traitement de signal.

- Formation de voies et filtrages associés en Sonar (64, 128, 256 chaînes simultanées).

- Filtrage spatial et temporel, balayage électronique, goniométrie etc ...

En général ces différents traitements concourent à un but commun et doivent donc être associés, ce qui se traduit par une tâche de synthèse (formation de Lobe par exemple), tâche qu'il faut effectuer le plus tard possible et le plus en aval possible.

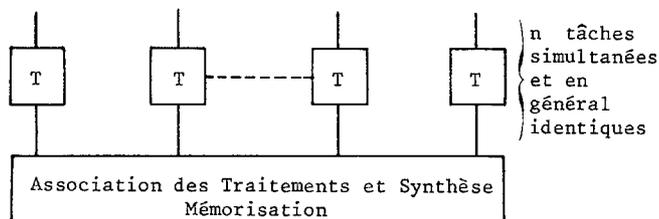


Figure 3

Traitements vectoriels avec simultanéité

Les processeurs de tableaux doivent donc permettre de contrôler  $n$  éléments identiques par une seule séquence d'instructions, en agissant sur de multiples flots de données.

### 2.12 - Traitements "vectoriels ou matriciels" sans simultanéité

Ces traitements sont éventuellement concurrents, c'est-à-dire effectués au cours de l'intervalle de temps considéré. Avec un processeur extrêmement rapide les tâches correspondantes peuvent donc parfois être traitées de manière séquentielle. Mais ces traitements étant vectoriels et donc indépendants, il est plus naturel de démultiplexer les tâches et de les exécuter en parallèle : le problème est ici plus simple qu'en 2.11 car la contrainte de simultanéité n'existe pas.

On peut noter par exemple sur la figure 4 que l'on a fait l'hypothèse que chacune des tâches  $T$  était exécutée deux fois moins rapidement qu'avec une organisation séquentielle dans le cas où l'on adopte une organisation parallèle.

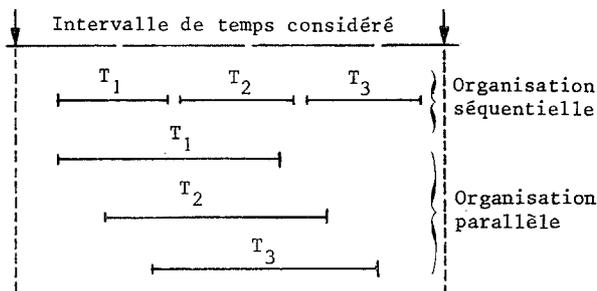


Figure 4

Traitements vectoriels sans simultanéité

Les exemples de traitements vectoriels sans simultanéité sont également très nombreux en traitement du signal. Le cas le plus typique est celui où l'on considère deux senseurs (deux antennes par exemple) physiquement indépendants dont les mesures sont associées au niveau du traitement de l'information. Il s'agit par exemple des couples :

- { Radar de poursuite en écartométrie
- { Télémètre Laser
- { Télémètre Laser
- { Ecartomètre infra-rouge
- { Sonar passif
- { Télémètre sonar etc ...

Multiplexer en amont est certainement la plus mauvaise méthode puisque les chaînes de traitement sont complètement indépendantes (sur le plan "spatial" et sur le plan temporel).

### 2.13 - Traitements "scalaires": parallélisme dans les traitements arithmétiques et logiques

Si l'on passe maintenant du niveau macroscopique au niveau microscopique, il existe dans chaque chaîne de traitement des tâches qui exigent d'effectuer des modules consistant en des opérations arithmétiques ou logiques, des opérations de mémorisation ou de lecture, des opérations d'entrée-sortie etc ... qui doivent être exécutées le plus rapidement possible (avec simultanéité ou non) pour que la tâche soit elle-même exécutée rapidement. L'accroissement des performances peut conduire à une exécution simultanée (dans des opérateurs spécialisés du processeur associé à la tâche) de modules qui ne sont pas simultanés par essence.

Ceci est d'ailleurs une solution classique dans certains ordinateurs de très haute performance dont l'organisation correspond à un processeur "multi-opérateur" pour les opérations arithmétiques et logiques et multiprocesseur (asymétrique ou non homogène) pour les entrées-sorties, les accès à la mémoire.

Le CDC 7600 (comme le CDC 6600 d'ailleurs) présente cette double particularité :

- Unité Arithmétique et Logique (UAL) "multi-opérateur" : deux opérateurs d'incrément, un additionneur, un additionneur double longueur, un multiplieur, un diviseur, un opérateur de décalage, un opérateur booléen, un opérateur de branchement, qui peuvent être utilisés simultanément.

- Organisation en multiprocesseur asymétrique pour les E/S et la mémoire. Dans le CDC 7600 il existe des processeurs périphériques (PP) spécialisés pour chacun de ces modules.

Il existe des machines de traitement de signal qui ont été construites selon ce type d'organisation.

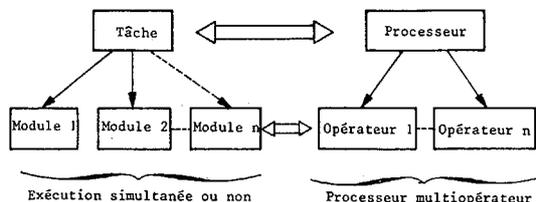


Figure 5  
Processeur multiopérateur

2.2 - Notion de traitement "pipe-line" (ou cascade)

Ici également le problème peut être considéré au niveau macroscopique (tâche, processeur ou multiprocesseur) ou microscopique (module, opérateur).

2.2.1 - Traitements pipe-line - tâches associées

Comme cela a été montré au paragraphe 1, les traitements envisagés appartiennent soit au domaine temporel (filtrage numérique récursif ou non), soit au domaine fréquentiel (Transformée de Fourier discrète). Si l'on admet que le parallélisme des traitements vectoriels est acquis, le problème est de savoir s'il est possible de dégager des structures pipe-line de traitement dans chacun des deux domaines.

a) Filtrage (numérique) en temps discret

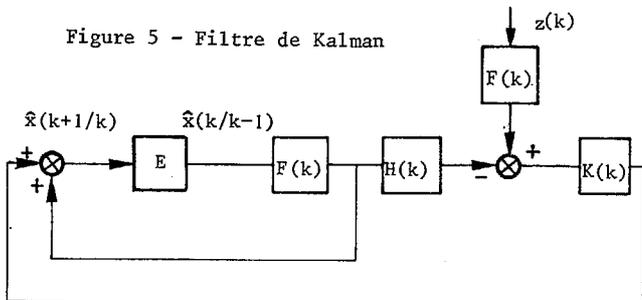
Nous n'envisagerons pour la clarté de l'exposé que le cas du filtrage linéaire optimal (Kalman-Bucy). Les équations matricielles du filtre sont de la forme (en représentation d'état) :

$$\hat{x}(k+1/k) = F(k) \hat{x}(k/k-1) + F(k)K(k) [z(k) - H(k)\hat{x}(k/k-1)] \quad (1)$$

$K(k)$  étant le gain de Kalman obtenu à partir d'une équation de Riccati que nous n'écrirons pas,  $F$  étant la matrice de dynamique et  $H$  la matrice de mesure,  $x$  l'état et  $z$  la mesure.

$\hat{x}$  est un vecteur d'ordre  $n$  et  $z$  un vecteur d'ordre  $m$  et la programmation de cette équation ressort dans une première approche du cas des traitements vectoriels simultanés analysé ci-dessus. L'utilisation de machines séquentielles programmées conduit donc à des résultats désastreux (cf. Appendice A et références [38], [39], [43], l'unité de temps étant la milliseconde pour un filtrage de WIENER (K asymptotique filtre stationnaire) et la seconde pour un filtrage de KALMAN pour un ordre moyen ( $n = 20$ ).

Figure 5 - Filtre de Kalman



Nous allons envisager dans ce qui suit le cas stationnaire pour simplifier l'exposé. La représentation interne de la figure 5 peut alors être transformée en représentation externe (ou de transfert)

$$\hat{X}(E) = E [I - F(I - K H)E]^{-1} F K Z(E) \quad (2)$$

avec  $E = e^{-Tp}$   $p$  étant la variable de Laplace.

Si l'on considère par exemple le cas monovariante où la grandeur filtrée ( $\hat{y}$ ) est l'observations  $z$ , il vient alors ( $m=1$ )

$$(3) \hat{y}(k) = H \hat{x}(k/k-1) \text{ et donc}$$

$$(4) \frac{Y(E)}{Z(E)} = G(E) = \frac{A(E)}{B(E)} = E H [I - F(I - KH)E]^{-1} F K$$

$G$  étant une fraction rationnelle en  $e^{-Tp}$  à coefficients réels. Au transfert  $G(E)$  d'ordre  $n$  correspond une équation aux différences elle-même d'ordre  $n$  qui exige donc  $n^2$  opérations d'addition-multiplication.

Le transfert  $G$  peut toujours être écrit sous une forme de produit de termes du second ordre.

$$G(E) = G_0 \prod_{i=1}^n \frac{1 + \alpha_{1i} E + \alpha_{2i} E^2}{1 + \beta_{1i} E + \beta_{2i} E^2} \quad (5)$$

Cette structure cascade correspond à  $n$  traitements  $T$  identiques mis en "pipe-line" (aux valeurs numériques près)

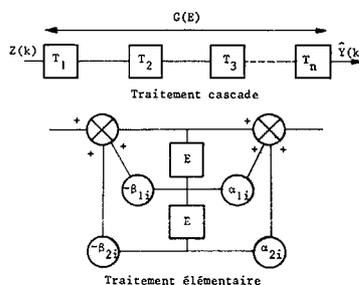


Figure 6 - Filtre linéaire stationnaire

La cellule élémentaire est très simple. Un seul opérateur de "multiplication-addition" peut d'ailleurs être utilisé en temps partagé pour les quatre opérations nécessaires dans une cellule élémentaire. Une réalisation en matériel spécifique et en technologie TTL [40] conduit par exemple à une multiplication  $16 \times 16$  en 300ns et à l'implémentation de 60 filtres à deux pôles et deux zéros avec une fréquence d'échantillonnage de 8 kHz par signal analogique (ou à l'implémentation de 15 filtres à huit pôles et huit zéros pour 15 signaux simultanés).

Les mêmes solutions peuvent naturellement être programmées ou microprogrammées. On constate qu'il est toujours possible d'appliquer une solution pipe-line au problème du filtrage numérique classique.

b) Transformée de Fourier discrète

La Transformée de Fourier Discrète (TFD) est définie par :

$$(6) X(p) = \sum_{n=0}^{N-1} x(n) \exp \left[ -\frac{2\pi j}{N} n p \right]$$

$n$  et  $p$  sont entiers,  $x(n)$  sont  $N$  valeurs discrètes,  $\alpha(p)$  est une fonction périodique de période  $N$ . Si  $x(n)$  est à valeurs réelles,  $X(p) = X^*(N-p)$  et l'expression (6) permet d'obtenir  $\frac{N}{2}$  valeurs indépendantes de la TFD.

Le calcul par (6) des  $N$  valeurs de la TFD nécessite  $N^2$  produits et  $N$  additions. La Transformée de Fourier Rapide ou TFR (Fast Fourier Transform FFT) est un algorithme, maintenant bien connu, et proposé par Cooley et Tukey en 1965 [45].

$$\text{En posant } W_N^q = \exp \left[ -\frac{2\pi j}{N} q \right]$$

$$y(m) = x(2m)$$

$$z(m) = x(2m+1) \quad m \in \left[ 0, \frac{N}{2} - 1 \right]$$



Il vient  $X(p) = Y(p) + W_N^p Z(p)$   
 (2)  $X(p + \frac{N}{2}) = Y(p) - W_N^p Z(p)$

Les relations montrent que l'on peut obtenir un TFD de N points à partir de deux TFD de  $\frac{N}{2}$  points en effectuant  $\frac{N}{2}$  produits et N additions. Chacun de ces TFD de  $\frac{N}{2}$  points peut être obtenu à partir de TFD de  $\frac{N}{4}$  points. Si M est une puissance de 2 ce processus itératif est reproduit  $\log_2 N$  fois et le nombre total d'opérations est  $N \log_2 N$  produits et  $N \log N$  additions. Pour  $N = 1024$  on gagne un facteur 200 sur le nombre de produits et 100 sur le nombre d'additions.

La séquence des opérations à effectuer pour l'algorithme de Cooley Tukey est représentée par le graphe de la figure pour  $N = 8$ .

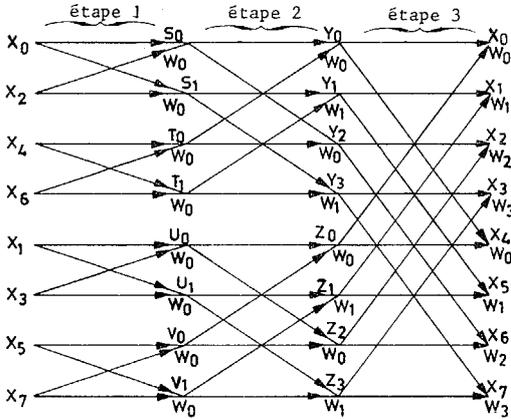


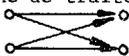
Figure 7  
 Graphe correspondant à l'algorithme de Cooley Tukey pour  $N = 8$

On constate sur la figure 7 que le graphe n'est pas régulier et que les quantités calculées ne sont pas disposées dans l'ordre naturel (trois étapes de calcul et quatre calculs de base par étape). Ceci n'est pas gênant lorsque l'algorithme est programmé sur un processeur utilisant des mémoires à accès aléatoire, en adoptant une réalisation de la FFT que l'on peut qualifier de séquentielle qui présente l'inconvénient complémentaire d'exiger une unité arithmétique et une mémoire rapides et donc coûteuses : pour  $N = 4096$  avec un temps d'exécution de 8 ms, l'unité arithmétique doit être capable d'effectuer quatre multiplications et six additions en 1/3 de microseconde.

Une première solution [47] consiste à modifier le graphe pour qu'il devienne régulier et permette la génération d'une séquence d'adresse identique à chaque itération et l'utilisation de mémoires circulaires.

Une deuxième solution consiste à utiliser un schéma de traitement de type pipe-line ou cascade [48]. Si l'on considère à nouveau le graphe de traitement de la figure 7, on peut réorganiser le graphe de manière que les échantillons de la transformée de Fourier soient délivrés dans l'ordre naturel (cf. figure 8)

Le graphe de traitement élémentaire (GTE) est de la forme



Si l'on considère que les deux premiers GTE de l'étape 1 ont été calculés (sur les données  $x_0, x_4$  et  $x_2, x_6$ ) les entrées nécessaires au calcul du premier GTE de la deuxième étape sont disponibles. De même après que trois GTE de la deuxième étape ont été calculés, les entrées nécessaires au calcul correspondant au premier GTE de

la troisième étape sont disponibles.

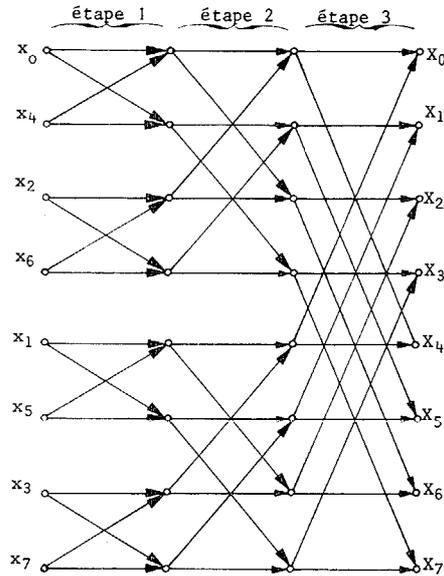


Figure 8  
 Algorithme de FFT de type "cascade" pour  $N = 8$

Le processeur de FFT peut donc commencer les calculs de l'étape 2 dès que les deux premiers GTE de l'étape 1 sont disponibles, et de l'étape 3 dès que les trois premiers GTE de l'étape 2 sont eux-mêmes disponibles. Le traitement est donc de type "pipe-line" avec plusieurs opérateurs identiques en cascade. Cette disposition minimise la vitesse à laquelle le processeur doit opérer. Pour  $N = 4096$  il y a  $\log_2(4096)$ , c'est-à-dire 12 étapes. Chaque unité arithmétique complexe (opérateur en cascade) devrait exécuter les opérations nécessaires en 4 microsecondes, c'est-à-dire 12 fois moins rapidement que dans le schéma séquentiel classique. Il est également possible d'utiliser l'UA complexe en temps partagé en augmentant sa vitesse (dans un rapport 4 par exemple) ce qui permet de ne conserver que 3 UA et non 12 dans l'exemple choisi.

2.22 - La notion de traitement "pipe-line" dans les traitements arithmétiques

Les organisations "cascade" peuvent être également appliquées au niveau des opérateurs élémentaires de l'Unité Arithmétique et logique. Un exemple caractéristique d'une telle organisation est l'addition en flottant de deux vecteurs de n éléments : sur un flot de données important les temps d'exécution peuvent être divisés par un facteur 4 [24].

De nombreuses machines utilisent des opérateurs de type cascade ; pour les ordinateurs non spécialisés l'IBM 360/91, le CDC 7600. Pour les processeurs spécialisés on peut citer :

- Le Texas Instrument Advanced Scientific Computer (ASC) dont le processeur central utilise des unités en cascade (au nombre maximal de 4 et dont le cycle de base est de 60 ns). Ce processeur permet de traiter d'autre part des entités vectorielles.

- Le CDC STAR 100 (SString ARray) dans lequel existent environ 30 opérateurs en cascade (cycle de base 40 ns) pour les opérations arithmétiques en flottant.

Il est bien évident que de telles structures posent des problèmes complexes d'efficacité ("pipe-line rempli") et de prise en compte des interruptions.



2.3 - Opérateurs élémentaires en traitement du signal - Unité Arithmétique de Traitement de Signal (UATS)

L'opération arithmétique de base qui doit être effectuée par un processeur de traitement de signal est la multiplication rapide sous la forme d'une opération "produit-somme" pour le traitement dans le domaine temporel et d'une opération "multiplication complexe" pour le domaine fréquentiel et la FFT. Ces opérations concernent toujours les blocs de données structurées en provenance des senseurs.

Les traitements qui doivent être pris en compte par l'Unité Arithmétique de Traitement de Signal (UATS) présentent d'autre part les caractéristiques suivantes :

- Algorithmes relativement simples.
- Opérations très répétitives.
- Précision demandée souvent faible.

Ce dernier point est très important et est très souvent ignoré. Dans la grande majorité des applications des mots de 16 bits (qui permettent un dynamique de 96 dB) suffisent. Dans certaines applications (formation de voies en sonar par exemple) le nombre de bits utiles est beaucoup plus faible et les opérateurs correspondants moins contraignants.

Pour que l'UATS permette à la fois les traitements dans les domaines temporel et fréquentiel, elle doit pouvoir exécuter l'opération suivante (avec adressage implicite) :

$$D_i \leftarrow \pm A_i^* B_i \pm C_i$$

i variant par exemple entre 1 et 4096 et A, B et C étant des nombres complexes.

Un multiplieur 16x16 de base est équivalent sur le plan de la complexité du hardware à quatre multiplieurs 16x16 ou à un multiplieur 32x 32 en virgule fixe.

La synthèse de l'UATS fait très largement appel aux techniques d'opérateurs pipe-line décrites ci-dessus et qui permettent une très grande rapidité d'exécution. D'après [18] en se basant sur une réalisation en technologie ECL avec une durée de microinstruction de 50 nanosecondes, une multiplication complexe peut être exécutée en 100 nanosecondes.

L'optimisation de l'UATS peut d'ailleurs reposer sur d'autres techniques. Il est possible en effet d'exploiter certaines aides câblées pour réaliser l'opération de multiplication de base.

L'accélération obtenue est due à l'association de circuits combinatoires à des organes séquentiels conduisant à des automates plus performants que les automates classiques. L'anticipation au niveau de certaines microinstructions permet d'optimiser les temps d'exécution.

3. - ORGANISATION DES MACHINES DE TRAITEMENT DE SIGNAL

3.1 - Définition d'une structure à multi-processeurs [24]

Une définition générale peut être élaborée en considérant simultanément des critères relatifs au logiciel et au "hardware":

a) Il existe deux ou plusieurs processeurs centraux (Central Processing Unit CPU) qui peuvent être, soit identiques (ou du moins de "possibilités" comparables) soit différents (systèmes asymétriques).

b) La mémoire principale de chaque processeur est

partagée et accessible par tous les processeurs. Il peut d'ailleurs exister, ou non, des mémoires spécifiques de chaque processeur et accessible par lui uniquement.

b) Les Entrées-Sorties sont partagées (canaux, unités de contrôle etc ...)

c) Il existe un système d'exploitation unique qui permet le contrôle global du logiciel et du hardware.

e) Il existe des interactions étendues tant au niveau du logiciel que du hardware.

- au niveau du logiciel dans l'exécution des tâches système,

- au niveau du programme dans la possibilité d'exécution d'un même programme par plusieurs processeurs et d'une tâche indépendante par un des processeurs qui n'exécute pas la tâche principale,

- au niveau des données et de leur implantation,

- au niveau des interruptions.

Les multi-processeurs ne doivent donc pas être confondus avec certaines organisations multicalculateurs qui ne répondent pas aux cinq critères précédents.

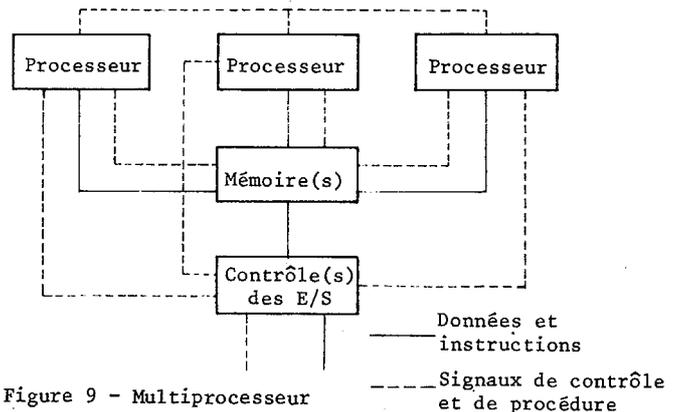


Figure 9 - Multiprocesseur

3.2 - Quelques éléments sur le développement des multi-processeurs et des systèmes à organisation parallèles

Les multi-processeurs se sont développés dès 1960 dans le domaine des applications de calcul scientifique ou de gestions et dans le domaine des applications militaires et aérospatiales. Leur développement a été surtout freiné par la difficulté d'élaboration de logiciel "système" bien adaptés à leurs possibilités.

3.2.1 - Applications de calcul scientifique

Il s'agit surtout de machines de taille importante parmi lesquelles on peut citer par exemple les ordinateurs suivants :

- Burroughs B5000 (Février 1963), B5500 (Novembre 1964) B5700 (Octobre 1970) B6700 (Juin 1971) B7700 (Février 1972).

- CDC 6600 (Septembre 1964) 7600 (1968) CYBER 72, 73, 74, 76 (1972).

- UNIVAC 1108 (Décembre 1965) et 1110 (1971).

- IBM 360/65 MP (Avril 1969) et 370/158 MP et 168 MP (1974).

3.2.2 - Domaine aérospatial et militaire

Si l'on considère, par exemple, les matériels produits aux Etats Unis, il est évidemment beaucoup plus délicat de disposer de listes exhaustives dans ce domaine. Il est cependant possible de considérer comme un exemple type l'UNIVAC AN|UYK-7 (Sperry Rand Corporation



qui produit d'ailleurs de "nombreux" types de multi-processeurs ou multiopérateurs). Ce calculateur comporte dans sa configuration la plus élaborée :

- Trois processeurs centraux.
- Quatre contrôleurs et quatre adaptateurs d'entrée sortie.
- 256 Kmots de 32 bits de mémoire à tore (8 accès partagés entre trois processeurs et deux contrôleurs d'E/S ou deux processeurs et quatre contrôleurs).
- Organisation multibus.

Les autres machines sont, soit plus complexes (Structure matricielle ou vectorielle), soit plus spécialisées (Structure pipe-line, machine multiopérateurs etc ...)

Constructeur	Type	Nombre de processeurs	Mémoire	E/S	Organisation	Performances	Applications
UNIVAC	ANVU K 7	1 à 3	256 Kmots de 32 bits 8 accès Cycle mémoire 1,5 µs	4 contrôleurs et 4 adaptateurs	Parallèle Multibus	Addition 1,3 µs Multiplication 10 µs	Advanced Missile System High Speed Attack Submarine SIGP
	ANVU K 20 (UNIVAC 1816) Version simplifiée de l'ANVU K 7	1	65 Kmots de 16 bits Cycle mémoire 750 ns Microprogrammation	1 contrôleur	Série DMA 2 bus	Addition 0,75 µs Multiplication 3,8 µs	
	1230		32 Kmots de 30 bits Mémoire hiérarchisée Mémoire à filtres magnétiques 250 mots	32 canaux	Parallèle	Addition 2 à 4 µs Multiplication 8 µs	Tenue de situation tactique Traitement de données
Sylvania Electronic Systems	FSP Advanced Signal Processor		15536 mots de 16 bits (8,12 ou 20 en option) Temps d'accès 66 ns Mémoire à tore de 6 Kmots		Parallèle	FFT sur 1024 Pts complexes 13,6 ms (programmé)	Traitement de Signal
Singer	SVC 2000		132 Kmots à tore Temps d'accès 1,2 µs Mémoire Scratch Pad LSI - mots de 16, 24 et 32 bits	69 canaux	Parallèle	Addition 3,1 µs Multiplication 6,6 µs	Traitement de Signal Commande et Contrôle
CDC Control Data	ALPHA Series	2 ou plus	16 X mémoire à tore (temps d'accès 1 - 3 µs) Mots de 36 bits 16 X mémoire à film magnétique	8 canaux par unité d'E/S	Parallèle matricielle	Addition (fil) 3,5 µs Multiplication 3,7 µs	Traitement de Signal Commande et Contrôle
Litton	Série 3000 (K 3000)	1	132 Kmots de 32 bits Temps d'accès 1,8 µs	64 canaux	Parallèle	Addition 4 µs Multiplication 10 à 30 µs	Traitement de Signal Commande et Contrôle

Figure 9

Principaux processeurs utilisés dans le domaine Aérospatial et Militaire aux Etats Unis

3.23 - Processeurs expérimentaux à organisation vectorielle - Processeurs de tableaux - Machines de laboratoire

Elles ont existé dès 1958 et la plus connue de ces machines est l'ILLIAC IV (Burroughs 1972) qui comporte quatre ensemble de 64 processeurs avec un processeur de contrôle par ensemble et un processeur central coordonnant les quatre processeurs de contrôle (hiérarchie de multiprocesseurs). Cette machine pourrait inverser une matrice (700x700) en une seconde.

On peut citer également STARAN (Goodyear Aerospace - 1972) qui a été spécifiée pour le tracking Radar de cibles multiples (multi-processeur vectoriel 32 ensembles de 256 processeurs et 64 Kbits), PEPE : (Parallel Ensemble of Processing Elements 1971) Bell Laboratoires pour le compte de l'US Army Advanced Ballistic Missile Defense Agency (un multi-processeur 7600 CDC utilisé comme organe de contrôle avec des unités complémentaires : corrélation, arithmétique, sortie) OMEN 60 (Sanders Associates) qui comporte un DEC POP 11 comme organe de contrôle et un processeur spécial (Unité arithmétique) et qui permet de traiter des problèmes d'algèbre linéaire, de FFT etc ...

Ces machines présentent pour la plupart une organisation vectorielle ou matricielle (processeurs de tableaux, ou processeurs de tableaux associatifs)

3.3 - Classification des multi-processeurs

En se basant sur le nombre et l'organisation

des flots de données et d'instruction et sur l'organisation des traitements (cf.paragraphe 2), on peut distinguer du point de vue fonctionnel quatre types de processeurs.

a) Monoprocesseur de base : Un seul cheminement pour les données et pour les instructions.

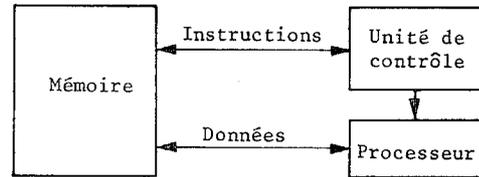


Figure 10 - Monoprocesseur de base

b) Multi-processeur de base : Organisation comportant des cheminements multiples pour les données et les instructions.

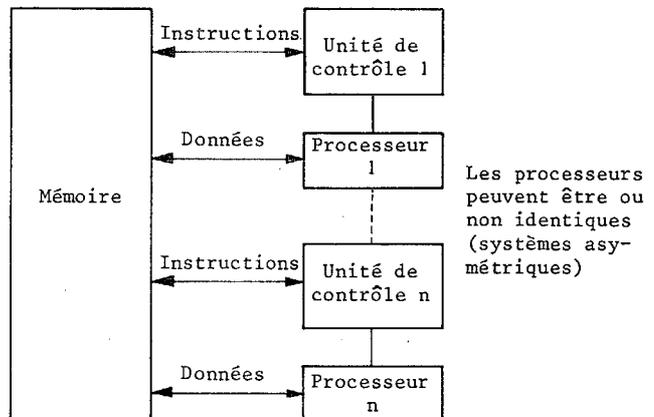


Figure 11 - Multiprocesseur de base

c) Multi-processeur avec centralisation des contrôles

Cheminements multiples pour les données et unique pour les instructions (processeurs vectoriels, processeurs de tableaux, processeurs associatifs)

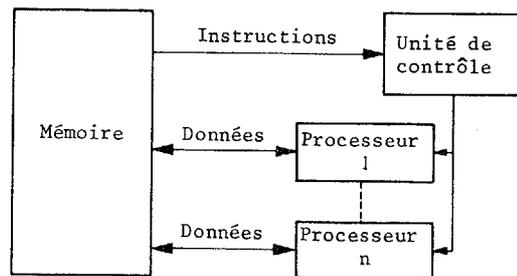


Figure 12

Multiprocesseur avec centralisation des contrôles

d) Multi-processeur uni ou multi-cascade (structure pipe-line)

3.4 - Organisation des structure à multi-processeurs

Les trois organisations les plus souvent adoptées sont :

- Le bus commun utilisé en temps partagé.
- Le multi bus associé à une mémoire à accès multiples.
- Les systèmes à connexion matricielle.

3.4.1 - Bus commun utilisé en temps partagé

Il n'existe pas de connexion établie en permanence entre les unités fonctionnelles. Le contrôle des transferts entre les modules de mémoire et les autres unités est accompli en utilisant des techniques de temps partagé ou de multiplexage.

Toutes les unités ou modules sont connectés en parallèle sur le bus (le standard des transferts pouvant être plus ou moins complexe et le bus pouvant être "série" ou "parallèle"). Le bus peut être mono-directionnel (figure 13a) ou tridirectionnel (figure 13b) Les interfaces et la logique de contrôle sont évidemment plus ou moins complexes, suivant que l'on adopte l'une ou l'autre des solutions.

Chaque bloc transféré sur le bus comporte les données à transférer et l'adresse de l'unité concernée. Il faut noter qu'il faut souvent une cinquantaine de fils (s'il s'agit d'un bus parallèle) pour transférer un mot de 16 bits.

Les principales caractéristiques d'un tel bus sont :

- La simplicité, bien que l'opération de multiplexage sur un bus unique puisse augmenter la complexité.
- La possibilité de rajouter simplement ou d'oter des unités fonctionnelles sur le bus. L'extension du système par additions d'unités risque par contre de dégrader les performances globales.
- La capacité limitée du système correspondant, résultant de la vitesse de transfert du bus.
- Les conséquences catastrophiques pour la disponibilité du système d'une panne du bus.
- L'efficacité faible du système multi-processeur correspondant (utilisation simultanée des différentes unités).

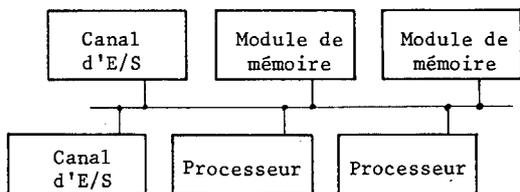


Figure 13a

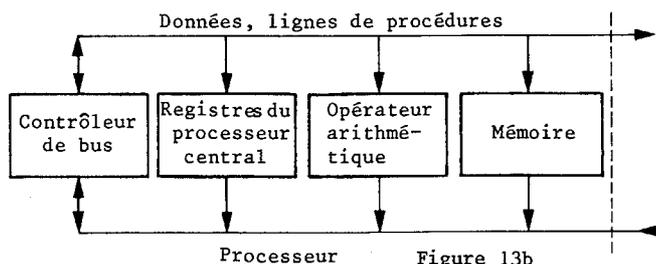


Figure 13b  
Organisation "bus-commun"

3.42 - Multibus

Cette organisation ne peut être adoptée que s'il existe des accès multiples sur les modules et des dispositifs permettant de résoudre les conflits qui se produisent quand plusieurs demandes d'accès au même module mémoire de processeurs ou d'unités d'E/S, apparaissent pendant un cycle mémoire. Certains dispositifs sophistiqués permettent un entrelacement des appels mémoire (entre modules), entrelacement qui assure une augmentation de la vitesse de transfert.

Les principales caractéristiques d'une telle organisation sont :

- La complexité plus grande des modules de mémoire
- Le fait que les processeurs n'exigent pas de circuits de commutation particuliers.
- Le faible coût de configurations "monoprocesseur"

utilisant les mêmes unités.

- La possibilité de vitesses de transfert élevées.
- Les limitations en taille maximale et en options de configurations inhérentes au principe même d'accès multiple mémoire (nombre et type d'accès disponible limités).
- Le grand nombre des câbles et de connecteurs exigés.

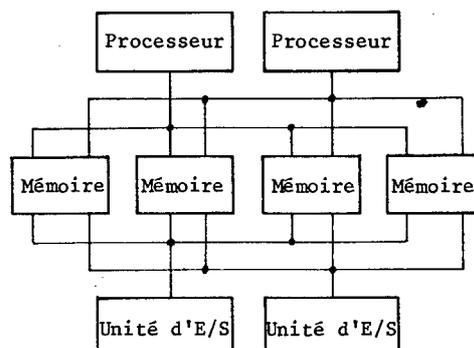


Figure 14  
Organisation multibus

Dans certains systèmes chaque processeur a de plus une mémoire "privée". Il est évident que cela pose des problèmes en cas de reconfiguration.

3.43 - Système à connexion matricielle

En multipliant les bus communs, on aboutit de manière naturelle à des organisations utilisant une matrice de commutation et dans lesquelles chaque module mémoire peut être connecté à chaque processeur ou à chaque unité d'E/S. La connexion n'est établie entre deux unités concernées par un transfert, que pendant la durée du transfert. Il s'agit en fait de techniques de commutation spatiale par opposition aux techniques de commutation temporelle du bus commun.

Les principales caractéristiques d'une telle organisation sont :

- La complexité et le coût du système d'interconnexion.
- La simplicité des unités fonctionnelles qui n'exigent aucun circuit de commutation spécifique.
- Le fait que ce type de configuration ne peut être utilisé que pour des multi-processeurs.
- La possibilité de vitesses de transfert élevées pour l'ensemble du système.
- Les possibilités importantes d'extension de la configuration par modification modulaire de la matrice de commutation.
- La fiabilité d'ensemble du système (liée à celle des organes de commutation qui peuvent être passifs, organisés en blocs etc ...)

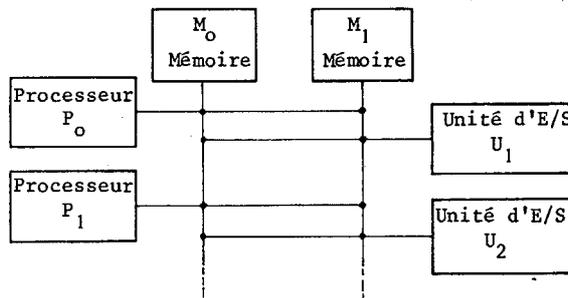


Figure 15 - Système à organisation matricielle



3.44 - Organisation des Entrée-Sortie et des interfaces

Dans ce qui précède, l'organisation des E/S et les interconnexions entre les unités de contrôle d'E/S et les différents modules ou dispositifs associés aux senseurs (coupleurs temps réel) ont été ignorées.

Tous les multi-processeurs utilisent en fait un des trois schémas d'interconnexion de base, suivant:

- Connexion matricielle avec commutation.
- Bus commun utilisé en temps partagé.
- Arborescences et schémas de connexion linéaires.

Nous n'insisteront pas sur les deux premiers points qui conduisent à des organisations de même type que celles envisagées en 3.31 et 3.33. Notons que pour l'organisation avec bus il est possible de définir un bus spécifique pour les E/S distinct du bus associé aux processeurs.

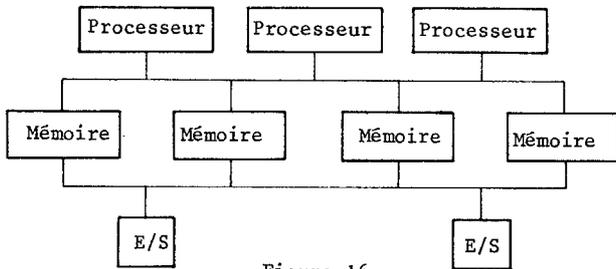


Figure 16.  
Bus d'E/S

Les organisations arborescentes sont très diverses, les multiplexages peuvent être effectués au niveau des unités de contrôle d'E/S ou des coupleurs temps réel.

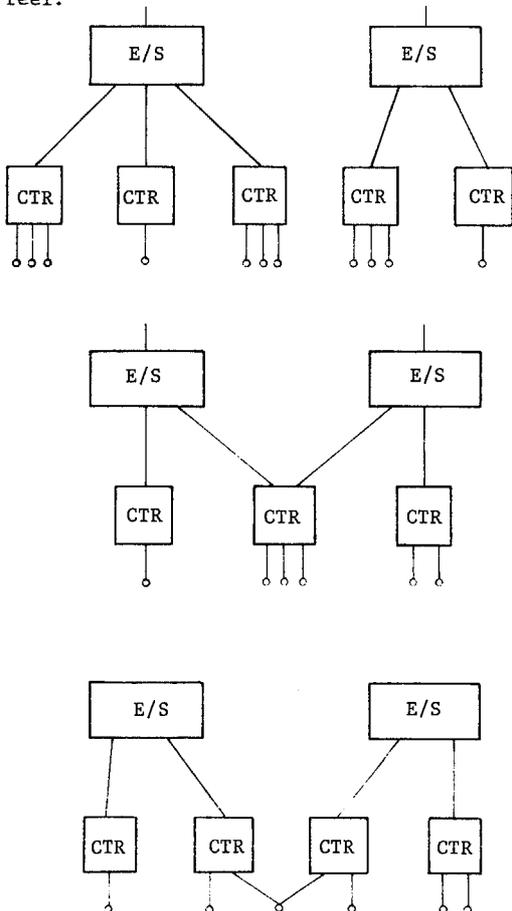


Figure 17 - E/S ; Organisation arborescente

3.5 - Microprogrammation et application aux processeurs de traitement de signal

3.51 - Avantages présentés par l'utilisation de la microprogrammation

En plus de la systématisation des circuits de commande, la microprogrammation permet de réaliser une synthèse "hardware-software" très importante lors de la conception du processeur spécialisé en traitement du signal. En effet :

- La simulation de la machine est aisée et permet la vérification complète des algorithmes. La même structure hardware peut correspondre à des calculateurs complètement différents pour l'utilisateur par simple remplacement des mémoires mortes : il est possible d'émuler des codes d'ordre extrêmement variés comportant des instructions spécialisées dans les problèmes de traitement du signal.

- La microprogrammation apporte d'autre part une économie appréciable en place mémoire et un gain important en rapidité d'exécution pour les opérations répétitives, ce qui est bien entendu le cas des traitements considérés.

Si l'on fait maintenant l'hypothèse que les fonctions arithmétiques de traitement de signal sont réalisées par une UATS (Unité Arithmétique de Traitement de Signal rapide), le problème global de tout système informatique associé est de faire en sorte que cette UATS soit toujours chargée et utilisée de manière optimale : ce problème est corrélé avec l'acquisition et le contrôle des entrées et de la structuration des données de telle sorte qu'elles seront efficacement prises en compte par l'UATS. Les fonctions caractéristiques associées sont :

- Le contrôle des E/S.
- Le multiplexage ou le démultiplexage.
- La mise en forme des données (formatage) et les conversions de format.
- Les mises à l'échelle.
- Les problèmes de buffer etc ...

Les avantages liés à l'utilisation d'un processeur de contrôle microprogrammé (PCM) plutôt qu'à celle d'un processeur câblé spécifique ont été évoqués ci-dessus. Il faut y adjoindre la possibilité de définition de macro instructions de traitement du signal qui peuvent être exploitées en mettant de manière adéquate l'UATS sous le contrôle d'un microprogramme. On peut définir par exemple les macroinstructions suivantes : filtrage récursif, produit de convolution, FFT, FFT inverse, corrélation, opérations matricielles etc ... Le jeu d'instructions est ainsi spécialisé et optimisé pour les traitements décrits au paragraphe 2.

3.52 - Structure du processeur de traitement de signal microprogrammé

Cette structure, inspirée des travaux décrits dans [18] est par exemple celle de la figure 18

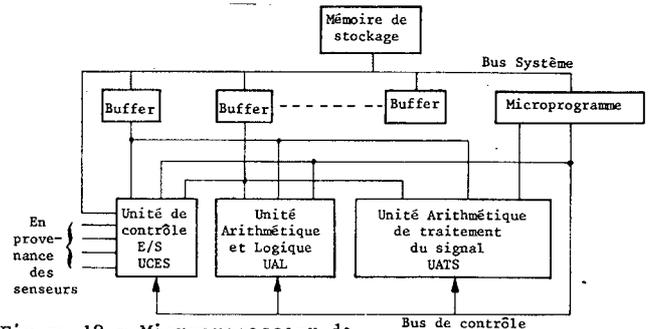


Figure 18 - Microprocesseur de traitement de signal



Le traitement des données est en général effectué par pages (ou par blocs) par l'UATS. La vitesse de transfert globale du système détermine les temps d'accès et de transfert entre les différentes mémoires.

Il faut noter qu'en traitement du signal les données et les programmes, pour une chaîne de traitement donnée, sont séquentiels par nature (même s'il existe des chaînes de traitements simultanées).

Les Buffers sont utilisés pour préparer le bloc de données à traiter par l'UATS qui fonctionne en technique pipe-line. Si l'on considère par exemple que les transferts sont effectués en un cycle élémentaire et que l'UATS effectue des opérations de base en deux cycles, il faut disposer de quatre buffers indépendants.

Les micro instructions doivent permettre, d'exécuter en parallèle des opérations d'Accès aux Buffers, de traitement de registres locaux ou de registres arithmétiques, des opérations arithmétiques et logiques, et de préparer l'adresse de la micro instruction suivante. Ces micro instructions font 32, 48 ou 64 bits : Ces tailles permettent des formats de contrôle beaucoup plus efficaces, surtout pour ce qui concerne les problèmes d'interface, que ceux obtenus dans les minicalculateurs.

En conclusion le PCM est un processeur de contrôle et de gestion d'interfaces. Ses caractéristiques sont :

- des buffers et des canaux très rapides,
- une organisation et une gestion efficace des interruptions,
- des interfaces simples,
- une microprogrammation avisée.

### 3.6 - Problèmes de logiciel

Nous envisagerons tout d'abord les problèmes posés par le logiciel de microprogrammation du PCM.

#### 3.6.1 - Logiciel du PCM

L'intérêt porté à un préprocesseur à organisation parallèle est surtout fonction du logiciel associé et de sa qualité. En effet, s'il est relativement aisé de développer le préprocesseur sous l'aspect matériel (et un certain nombre de constructeurs le font) toute la difficulté repose sur la richesse, l'étendue et la souplesse du logiciel permettant de l'utiliser, de le simuler, de préparer les microprogrammes, de les modifier et de les corriger, etc ...

Le logiciel associé doit comporter :

- Un assembleur,
- un simulateur de la machine,
- un implanteur de microprogrammes (éventuellement),
- un ensemble paramétrique de programmes permettant de décrire l'environnement temps réel et les entrées-sorties du système dans lequel est intégrée la machine.

Ce logiciel doit être implémenté sur un calculateur scientifique standard et doit être facilement portable.

Le simulateur d'assemblage doit permettre une écriture symbolique (variables, étiquettes, fonctions). Compte tenu des M opérateurs choisis, il doit de plus effectuer un certain nombre de vérifications :

- sur la syntaxe,
- sur la compatibilité des opérations effectuées simultanément.

Le simulateur de la machine doit exister

préalablement à l'élaboration du "matériel" de la machine et permettre de la concevoir de manière optimale et de faire des choix raisonnés quant à son organisation. Il constitue donc un élément essentiel des spécifications.

Une fois la machine construite, il doit permettre d'exécuter en simulation en temps relatif (simulateur d'exécution), les programmes assemblés. Le simulateur d'exécution doit être muni des aides à la mise au point permettant d'effectuer des traces des registres (total ou partiel) de suivre le déroulement du programme en cycle par cycle, opérateurs par opérateurs, modifier des constantes, etc ... Il est bien évident que l'utilisation de ce logiciel est complétée par une exécution sur le pré-processeur en mode pupitre (moyen d'accès à la mémoire de programme) avec visualisation des instructions et du contenu des registres en pas à pas. Mais cette dernière phase ne peut se suffire à elle-même et ce en raison de l'organisation même de la machine.

Le simulateur d'environnement et des E/S temps réel est élaboré par chaque utilisateur pour son sous-système spécifique. Ce que le constructeur doit fournir, ce sont par contre les outils de logiciel (programmes et instructions de nature "descriptive") permettant d'élaborer simplement cette simulation sous forme paramétrique.

Il est souhaitable que les simulateurs puissent être supportés par un système plus général. Ce système doit inclure une structure permettant d'écrire de façon simple des programmes de simulation (E/S par exemple).

La solution optimale pour l'ensemble de ces simulateurs consiste en une exécution conversationnelle à partir de terminaux.

#### 3.6.2 - Programmation et logiciel des multi-processeurs

Les différences entre le logiciel d'un mono-processeur multiprogrammé de troisième génération et celui d'un multi-processeur ne sont pas évidentes si l'on se place d'un point de vue macroscopique : en fait le point le plus marquant est celui qui concerne l'utilisation extensive effective du parallélisme résultant de l'organisation à processeurs multiples.

La reconnaissance et l'exploitation du parallélisme des traitements est souvent très complexe : le langage de programmation (ou le programmeur !) doit permettre d'identifier les tâches ou les modules qui peuvent être exécutées en parallèle et fournir les informations nécessaires au moniteur pour contrôler leur exécution.

Nous n'envisagerons pas dans le détail ces problèmes qui ne concernent que partiellement le traitement du signal.

Il existe trois organisations principales possibles :

- Maître esclave.
- Moniteur séparé et spécifique de chaque processeur.
- Traitement symétrique ou anonyme de tous les processeurs.

#### Organisation maître-esclave

Le superviseur "tourne" toujours dans l'un des processeurs (banalisé ou particularisé) qui est sélectionné par l'opérateur : il s'agit donc d'une organisation relativement figée.

Les caractéristiques d'une telle organisation sont



Les suivantes :

- Les routines du superviseur ne sont pas toutes réentrantes.
- Les tables sont associées à un seul processeur : il n'y a donc pas en général de conflit.
- Le calculateur esclave est en attente chaque fois que le "maître", trop lent, est trop chargé.
- Les pannes ont a priori des conséquences très graves.

Moniteur séparé pour chaque processeur - Chaque tâche est affecté à un processeur particulier qui opère de manière autonome et exécute son superviseur comme s'il était un monoprocesseur.

- Le code pour le superviseur doit être réentrant ou dupliqué dans chaque processeur.
- Chaque processeur a ses tables propres, sa gestion propre, des E/S, etc ...
- Une panne ne conduit pas à une situation désespérée.
- L'efficacité peut baisser si l'un des processeurs est très chargé alors que les autres sont disponibles.

Traitement symétrique ou anonyme de tous les processeurs

Le moniteur "flotte" [24] d'un processeur à un autre, chaque processeur pouvant traiter les tâches qui lui sont étroitement associées ou les tâches communes et en particulier les tâches asynchrones (interruptions d'E/S).

- Les conflits dans les demandes de ressources sont résolues par des affectations de priorités.
- Le code doit être réentrant.
- Des conflits dans les accès aux tables et aux données peuvent se produire puisque les superviseurs peuvent être actifs dans plusieurs processeurs simultanément.

Cette organisation qui présente beaucoup d'avantages est très difficile à implémenter.

### 3.7 - Fiabilité et Disponibilité

C'est un point extrêmement important puisque l'une des motivations qui ont conduit à l'introduction de l'informatique dans les dispositifs de traitement du signal est l'amélioration de la disponibilité.

Il est évident qu'un multiprocesseur possède de par sa structure une disponibilité bien meilleure que celle d'un mono-processeur : il est en effet possible de donner à cet aspect des performances des systèmes une importance plus grande en introduisant des éléments redondants ou dupliqués et une organisation de reconfiguration adéquate en utilisant des schémas d'interconnexion des différentes unités fonctionnelles permettant par commutation des organisations "complètes". Le tableau de la figure 19 montre par exemple le gain en disponibilité que l'on peut attendre pour les MTBF (temps moyen entre pannes) et MTTR (temps moyen pour réparer) donnés de la duplication d'un processeur associé à une tâche.

	Temps moyen pour réparer (heures)	TEMPS MOYEN ENTRE PANNES/MTBF					
		1000 heures		2000 heures		3000 heures	
		Disponibilité %	Hors service	Disponibilité %	Hors service	Disponibilité %	Hors service
Un calculateur	2	99,8	17,5 h	99,9	8,8 h	99,93	5,8 h
	5	99,5	43,8 h	99,75	21,8 h	99,83	14,5 h
	8	99,2	69,6 h	99,6	35 h	99,74	23 h
Deux calculateurs	2	99,9998	63 sec.	99,9999	31 sec.	99,9999	31 sec.
	5	99,9987	6,8 mn	99,9968	1,7 mn	99,9998	1 mn
	8	99,996	14 mn	99,9992	4 mn	99,9996	2 mn

Figure 19  
Disponibilité d'un ensemble informatique à monoprocesseurs dupliqués

La disponibilité est le pourcentage du temps pendant lequel le système est en service au moins partiel (au moins un processeur sur deux)

De nombreux travaux relatifs à des structures permettant (tolérant) des erreurs ou des pannes ont été effectués aux Etats Unis ou en France :

- Système SIRU (Strapped Down Inertial Reference Unit) du MIT (1969) dans lequel les processeurs et les mémoires sont doublés et exécutent en synchronisation les mêmes travaux.

- IL/ACGN du MIT qui comporte des unités fonctionnelles permettant de réallouer les travaux dans les différentes unités après une panne et de détecter des erreurs.

- STAR (Self Testing and Repairing System) du Jet Propulsion Laboratory dont le caractère de multiprocesseur est surtout exploité pour la détection des pannes et la remise en oeuvre des fonctions de traitement.

La disponibilité est une fonctionnelle de la fiabilité des différentes unités du système considéré et de leurs composants. Dans les systèmes monoprocesseur typés, la disponibilité du système est une fonction simple de la fiabilité d'ensemble ("produit" des fiabilités élémentaires) et du MTTR (Mean Time To Repair). Par contre, dans un multiprocesseur où de nombreuses unités identiques sont disponibles et pour lequel une certaine souplesse dans les procédures de reconfiguration est permise, le problème ne peut être résolu qu'en utilisant une unité de contrôle spéciale qui remplit les fonctions suivantes :

- Séparation du hardware du multiprocesseur en sous-systèmes indépendants.
- Mise "hors système" des unités qui sont en maintenance et en test sans détruire l'exécution des opérations du reste du système.
- Stockage et protection des informations les plus importantes (mots d'états, contenu des registres etc..) dans l'éventualité d'une panne d'un des processeurs centraux ou des contrôleurs d'E/S.
- Initialisation Automatique d'une séquence de remise en route et de reconfiguration après détection d'une erreur ou d'une panne, la procédure de reconfiguration pouvant faire intervenir un opérateur.

Un système de disponibilité élevé doit donc être modulaire, redondant, reconfigurable, et apte à détecter des erreurs ou des pannes et à en minimiser l'impact.

On peut trouver dans [24] un exemple de disponibilité composé de systèmes mono et multiprocesseurs. On considère une application qui exige un minimum de un processeur, six modules de mémoire et deux unités d'E/S. Sur la base d'hypothèses de disponibilité pour chacune de ces unités élémentaires, on compare les disponibilités de quatre organisations de systèmes (et les exigences en nombre d'unités élémentaires).

	Disponibilité des unités élémentaires	Monoprocesseur	Deux monoprocesseurs (duplication)	Trois monoprocesseurs	Multiprocesseur
Processeur	0,98	1	2	3	2
Mémoire	0,97	6	12	18	8
Unité d'E/S	0,90	2	4	6	3
Disponibilité globale du système		0,66	0,88	0,95	0,951

Figure 20  
Disponibilité composée de multiprocesseurs et de monoprocesseurs

On constate que pour atteindre le taux de disponibilité du multiprocesseur, il faut en fait trois monoprocesseurs. Le problème n'est cependant pas si simple car :



- il n'a pas été tenu compte dans le cas du multi-processeur du système d'interconnexion entre les unités fonctionnelles qui est certainement plus complexe et moins fiable que celui d'un monoprocesseur.

- Le coût marginal de ce système d'interconnexion peut être très important,

- et surtout le logiciel permettant d'exploiter un tel système est plus complexe, plus coûteux, plus délicat à tester et donc moins fiable que celui d'un monoprocesseur.

4. - EXPLOITATION ET VISUALISATION - DIALOGUE OPERATEUR

4.1 - Rôles des dispositifs de visualisation

Le dispositif de visualisation graphique (vidéo brute) ou synthétique est très souvent le seul organe de l'ensemble de traitement de signal qui permette :

- la synthèse d'Image,
- le dialogue opérateur processeur de traitement

La synthèse d'image [4] consiste à présenter aux opérateurs, généralement sur console graphique, de la manière la plus aisément lisible, les informations nécessaires à leur travail. En effet, et comme le fait remarquer très justement LEFAUDEX pour le Sonar, "les dispositifs de traitement de signal et de traitement de l'information associés ne sont pas dans leur finalité des machines à améliorer le rapport signal à bruit, mais des machines décisionnelles".

Le dialogue opérateur peut être assez complexe. Il comporte en effet :

- des fonctions d'aide au traitement du signal: choix de paramètres en temps réel en fonction de la présentation graphique de certains résultats de traitement, lissage manuel de signaux etc ...
- des fonctions de surveillance et de reconfiguration,
- des fonctions d'optimisation à partir des résultats d'identification paramétrique en temps réel.

4.2 - Analyse sommaire des procédés de visualisation et des principales caractéristiques des visualisations actuelles

4.21 - Procédés utilisés

Les solutions faisant intervenir la couleur commencent à intervenir plus souvent :

- a) Solutions à balayage télévision
  - télévision couleur à shadow mask ou à grille,
  - télévision bicanon utilisant les phosphores à pénétration,
  - télévision à tube à index.
- b) Solutions à balayage cavalier avec phosphore à pénétration : pour lesquels sont plus particulièrement étudiés les problèmes :
  - de commutation de couleurs,
  - du mode de balayage (purement aléatoire ou séquentiel avec utilisation d'une mémoire de masse).
- c) Autres solutions - (tubes à mémoire, projections, etc ...)

4.22 - Générateurs de l'image synthétique

Ils permettent en particulier la création des différents graphismes synthétiques élémentaires dont est formée l'image synthétique (caractères alphanumériques, symboles, canaux lumineux, vecteurs etc ...) à partir de leur code envoyé par un dispositif d'élaboration des images.

Les générateurs synthétiques peuvent être classés en diverses catégories suivant le type d'écran qu'ils animent (télévision ou cavalier) et le type de graphismes qu'ils réalisent.

4.23 - Générateurs d'image graphique brute (vidéo réelle)

Ils permettent en particulier la création d'images graphiques brutes constituant la visualisation de fonctions complexes d'une variable (qui peut être ou non le temps). Ces chaînes de visualisation d'images brutes peuvent être analysées en fonction de leurs performances, de leur dépendance plus ou moins grande vis-à-vis des chaînes vidéo synthétique etc ...

On peut constater en particulier le développement de méthodes consistant à utiliser le même tube qu'en balayage cavalier classique en imposant un balayage de type télévision au spot selon une direction et une modulation d'amplitude, selon la direction perpendiculaire, la valeur de cette modulation étant lue séquentiellement dans une mémoire de masse (MOS par exemple) dans laquelle est stockée l'image "latente" (il s'agit en fait d'une sorte de balayage cavalier-séquentiel).

4.24 - Analyse des dispositifs de commande et de dialogue

Il faut considérer ici les dispositifs et les procédures de commande et de dialogue :

a) Claviers :

Numériques ou alphanumériques, d'ordre, à libellés variables, "logiciels" avec désignation par photostyle, par boule ou par touche banalisée, etc ...

b) Moyens de désignation

Boule (matériel ou logiciel), crayon lumineux (photostyle), manche (Joystick), manivelle.

c) Moyens de poursuite

Boule, crayon, manivelle, etc ...; en poursuite : tracé dynamique d'une courbe f(t) sur l'écran.

4.25 - Dispositifs de couplage et d'élaboration des images

On distingue, d'une part le calculateur central de traitement ou de gestion de données, d'autre part le coupleur spécialisé associé au(x) tube(s) de visualisation.

Le coupleur spécialisé a un certain nombre de fonctions :

- codage analogique/numérique,
- Déport d'information et fonctions de couplage avec le calculateur central,
- élaboration des vidéo brute et synthétique (avec ou non séparation des chaînes),
- gestion d'ensemble, soit au niveau des liaisons avec le calculateur central (liaisons qui sont de type asynchrone et "lente" : modification des listes de visualisation) soit au niveau des liaisons avec le tube (rafraîchissement par exemple : liaison de type synchrone et "rapides") ou avec le clavier etc ...)

D'autre part, la structure et la technologie de ce coupleur conduisent à distinguer :

- les coupleurs microprogrammés,
- les coupleurs microprogrammés et programmés,
- les coupleurs à organisation parallèle etc ...
- les coupleurs et les structures de console permettant des reconfigurations partielles ou globales (par



INFORMATIQUE ET TRAITEMENT DU SIGNAL

des moyens de matériel ou de logiciel etc ...).

4.26 - Logiciel

Les différents types et niveaux de logiciel sont en général les suivants :

- a) Logiciel de base associé au coupleur, micro-ordres et micro-commandes, langage associé au coupleur quand il est programmé (assembleur, macro-assembleur etc ...).
- b) Logiciel de base associé au calculateur central pour un coupleur ou une famille de coupleurs donnés : handlers nécessaires (modes de liaison calculateur central, coupleur etc ...).
- c) Logiciel associé à la génération des images : Certains produits permettent de décrire et de préparer les images, soit sur le coupleur, soit sur le calculateur central ou sur un calculateur "compatible" qui peut être un calculateur scientifique très répandu.

Ces produits permettent en général de décrire

- des images "statiques" (ne comportant aucune courbe  $f(t)$  mais pouvant être modifiées par le calculateur central ou par l'opérateur (images interactives),
- des images "dynamiques" qui sont modifiées en un temps réel (tracé d'une courbe  $f(t)$  à partir de données provenant d'un capteur et codées en temps réel) et qui peuvent également donner lieu à un dialogue opérateur (images interactives).

Ces produits peuvent être d'un niveau de définition syntaxique et sémantique plus ou moins élaboré. Ils peuvent être facilement "portables" (écrits en FORTRAN par exemple).

APPENDICE I

FILTRAGE NUMERIQUE

Temps de calcul global pour un filtre de KALMAN (20 équations d'Etat, 9 équations de mesure)

	1974 Calculateur rapide ( $\mu$ sec.)	1968 Calculateur de rapidité moyenne ( $\mu$ sec.)	1965 Calculateur lent ( $\mu$ sec.)	1963 Calculateur très lent sans virgule flottante câblée ( $\mu$ sec.)
Multiplication	6	10	100	800
Division	7	15	130	1000
Addition	2	3	50	300
Mise en mémoire	2	2	8	120
Incrémentatation du registre d'indexe	1	2	8	80
Temps de calcul global (sec.)	0,9	1,4	11,8	97,5

Le temps de calcul global est le temps nécessaire pour l'exécution des équations d'évolution des covariances et du gain (équations d'extrapolation comprises) ainsi que des équations d'estimation et d'extrapolation pour l'état [38].

Dans cet exemple 3300 mots de 32 bits étaient nécessaires pour le stockage du programme (900 mots d'instructions) des différents coefficients, des résultats intermédiaires etc ...

Des résultats dus à GURA et BIERMAN [39] permettent d'évaluer les capacités en mémoire nécessaires pour l'implantation d'un Filtre de KALMAN (instructions du programme non comprises)

Encombrement Mémoire (instructions non comprises)			
n $\approx$ m			
n $\gg$ m	n $\approx$ 1	n $\gg$ 1	m $\gg$ n
2,5 n <sup>2</sup>	3,5 n(n + 1,3)	3,5 n <sup>2</sup>	m <sup>2</sup>

n étant la dimension du vecteur d'état

m étant la dimension du vecteur de mesure

Temps de calcul global et encombrement mémoire pour des filtres de KALMAN estimateurs de cinématique (Tracking Radar trois axes; échantillonnage T=160 ms)

Filtre stationnaire	Prédiction			Filtrage			Données			Total		
	n=9	m=3	n=21	m=3	n=9	m=3	n=21	m=3	n=9	m=3	n=21	m=3
Temps de calcul (m s)	0,270	0,630	0,168	0,378					0,438		1,008	
Encombrement mémoire (mots de bits)	72	168	54	126	51	171	177	465				

Filtre non stationnaire	Etat		Gain	Covariances		Total
	Prédiction	Filtrage		Prédiction	Evolution	
Equations d'évolution des covariances et du gain						
Temps de calcul (m s)	0,270	0,228	0,207	1,674	0,432	2,811
Encombrement mémoire (mots de bits)	72	84	60	459	135	810

Ces éléments sont relatifs à un calculateur UNIVAC 1230 US Q 20 B (sans virgule flottante câblée) dont les principales caractéristiques sont rappelées ci-dessous.

	Accès mémoire	Multiplication	Division	Addition	Cycle mémoire
Temps $\mu$ s	0,9	8 $\mu$ s	14 $\mu$ s	2 à 4 $\mu$ s	2 $\mu$ s

APPENDICE II

Transformée de Fourier discrète FFT  
Comparaison des performances de quelques processeurs effectuant la transformée de Fourier

La liste qui va suivre n'a pas la prétention d'être exhaustive. Elle permet uniquement de comparer sur quelques exemples caractéristiques les performances des différents types de Transformateurs de Fourier rapide.

Transformateur câblé

- \* Transformateur à mémoire circulaire CSF (1971)  
N = 1024 points complexes Temps de calcul = 5ms  
N = 2048 points réels Temps de calcul = 6ms
- \* Transformateur cascade Bell (1973)  
N = 4096  
Temps de calcul = 8ms

Transformateur programmé

- \* Transformateur utilisant un processeur à opérateurs parallèles (Transfert 100 ns, opérations A et L 100 ns, multiplication 16x16 bits 300 ns, division 1  $\mu$ s)  
N=1024 points réels Temps de calcul = 3,5ms

Transformateur microprogrammé

- \* Microprocesseur spécialisé (temps d'exécution d'un micro instruction 100 ou 200 ns; multiplication: 2  $\mu$ s)  
Organisation pipe-line des traitements  
N=1024 points Temps de calcul = 2,5ms
- \* Processeur microprogrammé Rapide (Japon)  
N=1024 points Temps de calcul = 3 ms

Exemple de traitements caractéristiques sur un processeur microprogrammé rapide (temps d'exécution d'une micro-instruction 100 ns)

Programme de recherche de n-m valeurs minimales dans un tableau de n = 50 valeurs.

Occupation mémoire : 40 micro-instructions

Temps d'exécution : 1 à 5 ms

Calcul d'une moyenne m et d'une variance  $\sigma^2$  sur  $n < 50$  valeurs.

Occupation mémoire : 20 micro-instructions

Temps d'exécution : de l'ordre de 150  $\mu$ s

Calcul de l'écart type  $\sigma$

Occupation mémoire : 30 micro-instructions

Temps d'exécution : 6  $\mu$ s (précision  $10^{-3}$ )

BIBLIOGRAPHIE

Ouvrages ou documents généraux

[1] E. JOSEPH "La Société câblée est-elle pour demain ?" "L'Informatique" Numéro de Juillet Août 1974

[2] ARTAUD "Orientation des recherches en programmation et Structure des machines" Note 931 CPM/ST du 4 Juin 1974 STCAN/DTCN Paris 1974

[3] H. MERMOZ "Note sur le traitement du signal et ses applications en particulier à la DSM" Laboratoire du Brusuc DCAN de Toulon 1974

[4] P. LEFAUDEUX "Informatique et Sonar" Réunion GRETSI du 4/10/74 MARCOUSSIS (CIT-ALCATEL)

[5] F. de COULON "Méthodes actuelles de traitement des signaux" EPFL Lausanne NT Nr 7/1973

[6] G. ROUCAIROL et A. WIDORY "Programmes séquentiels et parallélisme" Revue Française d'Automatique, Informatique et Recherche Opérationnelle Paris Juin 1973

[7] A.V. OPPENHEIM "Digital Signal Processing" MIT Research Lab. of Electronics Février 1974

[8] "Signal Processing" Proceedings of the NATO Advanced Study Institute on Signal Processing - Edited by J.W.R. GRIFFITHS and P.L. STOCKLIN Academic Press New York 1973

[9] "Digital Signal Processing" and "Literature in Digital Signal Processing" - Edited by H.D. HELMS and L.R. RABINER IEEE Press New York 1973

[10] B. GOLD and Ch.M. RADER "Digital Processing of Signals" Mc Graw Hill New York 1969

Micro-processeurs, mini-ordinateurs, microprogrammation

[1] Y. SARRAZIN "Microprocesseurs : évolution ou révolution" AZI n° 33 Janvier 1975

[12] J. PAJUS "L'évolution des minicalculateurs" L'Onde Electrique vol 51 fasc 11 Décembre 1971

[13] D.J. THEIS "Microprocessor and Microcomputer Survey" Datamation December 1974

[14] H. FALK "The Microprocessor : Jack of all trades" IEEE Spectrum November 1974

[15] D. HOLLINGWORTH "Minicomputers : A Review of Current Technology Systems and Applications" Rand Corporation AD 783 316 July 1973 Santa Monica USA

[16] B. SOUCEK "Minicomputers in Data Processing and Simulation" WILEY and Sons 1972

[17] J.P. BOUNOT "Micro Informatique Systèmes répartis - La fin de l'économie d'échelle" L'Informatique Mai 1974

Processeurs de traitement de signal

[18] Y.S. WU "Architectural Considerations of a Signal Processor under Microprogram Control" Spring Joint Conference USA 1972

[19] Y.S. WU and G.L. KRATZ "Microprogrammed Interface Processor (MIP) and its Application to phased array radar" Tech Note Spring Joint Computer Conference USA May 1971

[20] G.D. HORNBuckle and E.I. ANCONA "The LX1 Micro-Processor and its Application to real Time Signal Processing" IEEE Transactions on Computers Vol C-19 Number 8 August 1970

[21] P.H. SWAIN "Implementation and Evaluation of ILLIAC IV Algorithms for Multispectral Image Processing" NASA CR 139 688 LARS.T. 16 III 273 July 1974

[22] H.S. STONE "Parallel Triagonal Equation Solver" NASA TM.X. 62370 April 1974

[23] W.R. SMITH et al "AN/UY K 17 (XB-1) Signal Processing Element Architecture" Naval Research Lab Washington NRL 7704 June 1974

Multiprocesseurs - Traitement parallèle

[24] COMTRE CORPORATION "Multiprocessors and Parallel Processing" P.H. ENSLOW Jr Editor - J.WILEY New York 1974

[25] "Information Processing 1974 - Proceedings of IFIP Congress 74" J.L. ROSENFELD Editor American Elsevier Publishing Company New York 1974

[26] S.E. ELMAGHRABY and A.N. ELSHAFEI "The Scheduling of Jobs on Parallel Processors : A Survey and Annotated Bibliography" North Carolina State Univ. Raleigh Operations Research - 23 p (NCSU-OR-97 ; AD-786 314/5WC) September 1974

[27] R.D. ARNOLD "Multi Associative Processor Systemes Architecture" Colorado Univ. 59 p (CU-CS-051-74 ; PB 236339/8WC) August 1974

[28] H. IIZUKA ; K. FUJII ; T. YUBA ; T. SHIMADA "A Survey and Evaluation of Innovative Computer Architectures" CIRC 179 Electrotechnical Lab 115 p (N74 29537/9WC) Tokyo Feb. 1974

[29] S.E. ORCUTT "A Novel Parallel Computer Architecture and Some Applications" STCAN-CS-74-430 TR-71 ; 47 p ; (PB-234-513/OWC) May 1974

[30] C.A. ELLIS and G.J. NUTT "Preliminary Thoughts on Degrees of Security in Multiprocessor Systems" CU-CS-03674 ; 13 p (PB 235 292/OWC) January 1974

[31] K.V. SASTRY and RY. KAIN "Markovian Models for Performance Analysis of Multiprocessor Systems" NSF-OCA-GJ 32504 TR-1 44 p (PB-235-843/OWC) January 1974

[32] L.C. HOBBS et al eds "Parallel Processing Systems Technologies and Applications" Spartan USA 1970

[33] AIAA Computer Network Systems Conference Huntsville Ala. USA - New York Am. Inst. Aeronaut Astronaut. 1973

[34] "Hiérarchie de mémoires dans les multiprocesseurs" Contrat de recherche SESORI n° 74 127 CCI Grenoble (Mr Bennett) 1974

[35] "Méthodes et techniques associatives dans les multiprocesseurs" Contrats de recherche SESORI n° 74 010, 74 014, 74 148 Université de Rennes (Mr BOULAYE) 1974

[36] "Architecture de Calculateur à contrôle décentralisé exploitant le concept d'assignation unique" Contrat SESORI n° 74 167 ONERA/CERT (Mr SYRE) 1974

[37] Etude d'Architecture "Matériel" et des processeurs associés - Contrat SESORI n° 74 185 CII (Mr RECOQUE) 1974

Filtrage linéaire optimal - Filtrage de Kalman

Filtrage récursif

[38] A. GELB "Applied Optimal Estimation" The MIT Press Cambridge Massachusetts 1974

[39] I.A. GURA and A.B. BIERMAN "On Computational Efficiency of Linear Filtering Algorithms" Automatica Vol 7 pp 299-314 1971

[40] R.A. GABEL "A Parallel Arithmetic Hardware Structure for Recursive Digital Filtering" IEEE Transactions on Acoustics, Speech and Signal Vol ASSP 22 n° 4 August 1974



- 
- [41] T.A BRUBAKER et al "Hardware Considerations for the Implementation of a Parallel Non Linear Filter" Colorado Univ. AFORS-TR-73-2206-1973
- [42] R. STEPHEN Mc REYNOLDS "Parallel Filtering and Smoothing Algorithms" IEEE Transactions on Automatic Control October 1974
- [43] P. FAURRE, L. CAMBERLIN, M. VIEVILLE "Systèmes de Navigation hybrides optimaux" Troisième Colloque GRETSI Nice 1971
- [44] E.T.S TSE et al "Parallel Estimation for Non Linear Air Force Problems" Air Force Office of Scientific Research TR 73 044 T USA February 1973  
Transformée de Fourier Rapide
- [45] J.W COOLEY and J.W TUKEY "An Algorithm for the Machine Calculation of Complex Fourier Series" Math of Comput, vol 19, April 1965
- [46] J.C CONSTANTIN, P. DELEPINE et J.L VERNET "Transformateur de Fourier Rapide à mémoires circulantes" Troisième Colloque GRETSI Nice Juin 1971
- [47] P. DELEPINE "Description et Applications de l'Analyseur FFT-TSM 4040 Revue Technique CFTH/CSF vol 5 n° 3 Septembre 1973
- [48] P.S FUSS and J.C STUART "Making Fast Fourier Transform Really Fast" Bell Laboratories Record February 1973