NEUVIEME COLLOQUE SUR LE TRAITEMENT DU
SIGNAL ET SES APPLICATIONS

NICE du 16 au 20 MAI 1983

ANALYSE DES DIFFERENTES ARCHITECTURES ENVISAGEABLES POUR UN DISPOSITIF DE TRAITEMENT RAPIDE

CRITERES DE CHOIX D'UNE CLASSE D'ARCHITECTURE

C.A. BOZZO

Compagnie de Signaux et d'Entreprises Electriques
17, Place Etienne Pernet - 75 738 PARIS

RESUME

Les besoins dans le domaine du traitement numérique rapide (pour le traitement du signal en particulier) ont conduit au développement d'architectures associant des processeurs et des opérateurs élémentaires selon des modalités qui permettent de distinguer différentes classes de dispositifs dont la finalité est d'obtenir des structures (de traitement) à haut parallélisme intrinsèque.

L'évolution remarquable des outils techniques et technologiques disponibles dans le domaine de la conception et de la synthèse des circuits VLSI à considérablement modifié la difinition des contraintes relatives aux architectures de traitement numérique rapide : il est désormais possible de développer des structures de traitement comportant un nombre très important de processeurs ou d'opérateurs (plusieurs centaines ou plusieurs milliers).

La communication proposée, a pour finalité de présenter les évolutions existant dans le domaine des SHPI en envisageant les réponses aux trois questions suivantes :

- quelles sont les architectures optimales pour des clauses spécifiques de problèmes ?
- quels sont les réseaux d'interconnexion et de communication des plus efficaces entre les différentes cellules ?
- quelles sont les méthodes d'analyse d'algorithmes, de représentation et de programmation de ces algorithmes sur les structures hautement parallèles?

On envisagera en particulier :

- les unités fonctionnelles multiples et les opérateurs systoliques
- les architectures de traitement à flux de données
- les dispositifs de communication suivants : Crosbar, réseaux en anneau, réseaux systoliques, architectures cubiques, etc...

SUMMARY

In many arrays of computer application new techniques for using very large numbers of computing elements to solve important problems have been developed recently.

Advances in the design and fabrication of VLSI circuits will soon make it feasible to implement processors consisting of tens or even hundreds of thousands of computing elements.

This article surveys three major questions :

- what are the optimal architectures for specifie classes of problems ?
- what are the most efficient interconnection networks on which to host the communication amony macrocells ?
- what are the most effective methods for algorithm disign and programming of highly parallel structures ?

This article examines some of the developments in

- multiples special-purpose functional units
- Dataflow architectures
- intercommunication devices : Crossbar, systolic arrays, ring networks, Cube-connected cycles etc..

ANALYSE DES DIFFERENTES ARCHITECTURES ENVISAGEABLES
POUR UN DISPOSITIF DE TRAITEMENT RAPIDE

CRITERES DE CHOIX D'UNE CLASSE D'ARCHITECTURE

1. - QUELQUES REFLEXIONS RELATIVES AUX PROBLEMES DE SYNTHESE DE DISPOSITIFS DE TRAITEMENT RAPIDE

1.1 - Caractérisation d'une structure de processeurs élémentaires à haut degré de parallélisme intrinsèque

Les besoins dans le domaine du traitement numérique rapide ont conduit au développement d'architectures associant des processeurs et des opérateurs élémentaires selon des modalités qui permettent de distinguer différentes classes de dispositifs dont la finalité est d'obtenir des structures (de traitement) à haut parallélisme intrinsèque (SHPI).

Une structure hautement parallèle (SHPI) présente les trois caractéristiques principales suivantes :

a) Elle est composée d'un grand nombre d'éléments de traitement et de calcul (cellules : opérateurs et processeurs) qui peuvent être hétérogènes.

b) Le nombre de ces éléments ou cellules est à priori extensible avec un coût de matériel (hardware) qui croît linéairement (ou du moins un peu plus rapidement que selon une loi linéaire) et ce, compte tenu de performances (au sens de la vitesse de traitement) qui doivent croître de façon sensiblement linéaire (ou du moins un peu plus lentement que selon une loi linéaire).

c) Elle peut être utilisée pour résoudre un problème unique à la fois

La synthèse de structures hautement parallèle conduit à envisager les trois principales questions suivantes :

a) Quelles sont les architectures optimales pour des classes spécifiques de problèmes et existe-t-il une méthode permettant de déterminer le profit marginal obtenu avec ces structures adaptées à des classes spécifiques de problèmes ? (par opposition à des structures de caractéristiques plus générales).

b) Quels sont les réseaux d'interconnexion* les plus efficaces (en termes de capacité de communication, de complexité du matériel et du logiciel, et de coût) entre les différentes cellules ?

c) Quelles sont les méthodes d'analyse d'algorithmes, de représentation et de programmation de ces algorithmes sur des structures hautement parallèles ?

1.2 - Limitations des capacités de traitement et des capacités d'accès

Les tâches de calcul et de traitement peuvent être classées en deux grandes catégories :

. les problèmes à "limitation par la capacité de calcul" (problèmes LCC)

. les problèmes à "limitation par la capacité d'entrées/sorties" (ou d'accès : problèmes LCA)

1.3 - Analyse de deux catégories de problèmes LCC

On peut également distinguer, parmi les problèmes LCC de calcul scientifique et technique, deux grandes catégories :

Les problèmes appartenant à la première catégorie sont ceux qui sont beaucoup trop "difficiles" pour être traités sur les machines existant actuellement : de façon générale, on peut caractériser ces problèmes par des équations aux dérivés partielles (EDP) non linéaires dans un espace 3 D.

Les problèmes appartenant à la deuxième catégorie peuvent être traités sur des machines "classiques", mais dans des conditions d'efficacité qui les rendent difficiles

Si l'on considère cette deuxième catégorie de problèmes, les solutions envisageables consistent à définir, développer et mettre en oeuvre des architectures de traitement nouvelles, et donc des systèmes de communication et de connexion entre processeurs (et opérateurs) efficaces. Les performances désirées ne peuvent, en effet, être obtenues dans la pratique en adoptant des moyens de traitement classiques regroupés en réseaux complexes ou en exploitant l'amélioration des qualités technologiques (fréquence d'horloge, bande passante des mémoires, etc) des composants de traitement (qui est, certes, importante mais n'est pas suffisante, du moins dans le domaine numérique).

Ces problèmes d'architecture sont complexes et souvent difficiles à appréhender, compte tenu du poids respectif de chacun des facteurs (performances des opérateurs, des mémoires, des dispositifs de communication, des entrées/sorties, etc). Il est cependant certain qu'un système de communication suffisamment souple et supportant des unités fonctionnelles standardisées, sous forme de composants VLSI spécifiques (modules FFT, algèbre linéaire, générateur de fonctions, etc) apporterait des améliorations notables aux performances des dispositifs de traitement rapide actuels.

Ces VLSI fonctionnels (que l'on retrouve, par exemple, dans le plan Américain VHSIC) peuvent, en effet, être considérés comme des modules (sous forme de matériels) remplaçant les sous-programmes d'une bibliothèque scientifique classique (sous forme de logiciels).

Les possibilités d'association de ces modules ou cellules constituent donc le noeud du problème.

On envisage essentiellement, dans ce qui suit, les problèmes appartenant à cette deuxième catégorie : bien que les dispositifs de traitement correspondants puissent parfois être exploités pour traiter des problèmes de la première catégorie, la méthode adoptée consiste à optimiser les développements effectués, de façon à traiter dans les meilleures conditions (au sens de critères à définir : performances, volume d'électronique, consommation, complexité et adaptativité des structures etc) les problèmes qui appartiennent essentiellement au domaine du traitement du signal au sens large (incluant l'analyse et la synthèse d'images, etc...).

1.4 - Parallélisme et problèmes de communication

Il existe, à priori, deux approches pour concevoir et développer un système de traitement et de calcul rapide. La première consiste à utiliser des composants très rapides et une architecture classique, la seconde à exploiter la simultanéité existant de façon intrinsèque dans certaines tâches (structures à haut parallélisme intrinsèque).

On peut appliquer le parallélisme de façon massive si l'algorithme a été conçu et étudié pour permettre l'introduction systématique de structures de type "cascade" et du multitraitement. Cependant, quand un nombre important d'éléments de traitement travaillent en simultanéité, les tâches de coordination et de communication acquièrent une grande importance et deviennent complexes, ce qui conduit donc, en général, à une diminution de "l'efficacité" de l'architecture adoptée.

On cherche donc à développer des algorithmes qui correspondent à un degré élevé de simultanéité, mais qui permettent également d'utiliser des dispositifs de régulation et de contrôle peu complexes, normalisés et répétitifs, de façon à autoriser une "implémentation" efficace et peu coûteuse.

ANALYSE DES DIFFERENTES ARCHITECTURES ENVISAGEABLES
POUR UN DISPOSITIF DE TRAITEMENT RAPIDE

CRITERES DE CHOIX D'UNE CLASSE D'ARCHITECTURES

1.5 - Poids respectif des problèmes de traitement et des problèmes d'accès - objectifs de performances

Un dispositif de traitement rapide reçoit de façon classique ses données et délivre ses résultats à travers un (ou plusieurs) "hôte (s)".

L'hôte, dans ce contexte, peut être un calculateur, une mémoire, un dispositif "temps réel", etc...
En matière de performances, l'objectif d'un dispositif de traitement rapide doit être une vitesse de traitement compatible avec la capacité d'entrées/sorties de l'hôte ("largeur de bande", débit instantané et moyen, etc).
Une grande vitesse de traitement n'est donc pas une fin en soi. Seul le débit global d'accès et de traitement a une signification (notion de "throughput").

1.6 - Opérateurs dans une structure à haut degré de parallélisme interne

Définition de la terminologie :

Le choix du "niveau" auquel s'effectue le découpage en cellules indivisibles est, bien entendu, fondamental quant aux conséquences qu'il implique sur la structure et l'architecture des réseaux et des dispositifs d'interconnexion et de communication.
On distinguera, dans ce qui suit les niveaux de décomposition suivants :

. microcellule

opérateur ou fonction élémentaire (OE), constitué d'un certain nombre d'éléments qui peuvent être des portes, des points mémoire, etc...

. macrocellule

juxtaposition de quelques microcellules (quelques unités ou, au maximum, quelques dizaines), permettant de réaliser un opérateur de calcul ou de traitement (OT)

. processeur élémentaire (PE)

ensemble de macrocellules

. processeur (PR)

ensemble de processeurs élémentaires.

2. - ANALYSE DE LA STRUCTURE D'UN PROBLEME DE TRAITEMENT - REPRESENTATION, DETECTION ET IMPLEMENTATION DU PARALLELISME

2.1 - Introduction - Sensibilité des performances vis-à-vis de la formulation du problème envisagé

Le problème de représentation s'est posé de façon aiguë avec le développement des méthodes de calcul parallèle et de calcul distribué. Les difficultés associées à la représentation des opérations de traitement, représentation devant permettre d'élaborer des structures conduisant à un haut niveau de parallélisme, ne se présentent que dans le cadre des architectures associées à un système distribué : la variété des situations rencontrées résulte de la variété même des schémas d'interconnexion possibles entre opérateurs et la diversité des dispositifs matériels correspondants.

2.2 - Les problèmes de dépendance et les différents types de dépendances

2.21 - Analyse des différents types de dépendances

Il existe 3 types de dépendances :

. dépendances relatives aux données

. dépendances relatives aux contrôles

. dépendances relatives aux ressources

Les deux premiers types de dépendances interviennent dans le domaine de la formulation et de la représentation des algorithmes, le troisième appartient par essence au domaine du matériel et des architectures de traitement.

Dépendances relatives aux données

Elles se subdivisent elles-mêmes en 3 catégories et elles assurent, en fait, que les valeurs, dont le calcul est projeté, sont bien exploitées dans une phase de traitement.

- Dépendances relatives au flux (de données) :
- Dépendances relatives aux sorties :
- Antidépendances :

Dépendances relatives aux contrôles

Les différents types de dépendances varient selon la syntaxe envisagée pour la description de l'algorithme.

Dépendances relatives aux ressources

Ces dépendances sont rencontrées quand sont introduites les contraintes d'architecture relatives à une structure de processeurs élémentaires particulière.

2.22 - Observation des dépendances

Un algorithme donné peut comporter certaines dépendances intrinsèques vis-à-vis des données. C'est le cas, par exemple, de certains calculs itératifs, pour lesquels une valeur doit être calculée avant qu'elle puisse être utilisée. Les algorithmes présentent donc des dépendances de nature séquentielle plus ou moins marquées et le choix effectué n'est pas indifférent.

En conclusion, les hypothèses de base, quant aux structures de données et de contrôle et aux dépendances vis-à-vis des ressources sont toujours, en fait, "gravées" dans le hardware des unités des contrôles et des dispositifs de communication.

2.3 - Algorithme et espace de représentation

2.31 - Espace de représentation

Un algorithme spécifie les tâches, composées de déclarations dans un langage de haut niveau, tâches qui sont représentées sous forme d'instructions unitaires qui conduisent à des transitions d'état dans la cellule de traitement envisagée.

Ces 5 niveaux constituent l'interprétation (ou modèle) hiérarchique du problème considéré.

Chaque niveau représente le même objet (programme) de différentes façons : les mêmes transitions et les mêmes ressources sont sollicitées, mais les codages correspondant à ces demandes sont différents.

2.32 - Représentation et détection du parallélisme

Le parallélisme peut se présenter dans une représentation, selon quatre variétés différentes :

Parallélisme explicite

Parallélisme localement détectable

Parallélisme globalement détectable

. Parallélisme non détectable :

Il s'agit, malheureusement, d'un cas qui se présente de façon courante dans la pratique. En effet, la détermination du degré de parallélisme d'un programme est, en général, un problème de nature indéterminée.

Ce dernier cas se produit en particulier quand il existe une dépendance vis-à-vis des données ou quand la représentation adoptée est "déficiente".

On peut, imaginer des méthodes d'analyse exploitant en particulier la théorie des graphes et permettant de "spécifier" le parallélisme. Les graphes associés aux algorithmes permettent de traduire de façon non ambiguë les dépendances (graphe de dépendances) existant au niveau de la formulation de l'algorithme, dépendances qui sont relatives aux données, aux contrôles et aux ressources.

La méthode proposée consiste donc, une fois le graphe de dépendances obtenu (sous une forme normalisée à spécifier), à transformer et modifier certains arcs et réduire certains sous-ensembles du graphe en noeuds de haut niveau correspondant en général à des macrocellules qui sont alors définies comme des unités fonctionnelles globales (en général spécialisées).

Le vecteur des macrocellules (et donc des fonctionnalités) disponibles peut par exemple être considéré comme une donnée préalable. Cette méthode d'implémentation directe assure que le dispositif de traitement permet de réaliser la plupart des structures de dépendance du graphe d'origine, ces structures ayant été transformées et modifiées pour tenir compte des contraintes du SHPI.

3. - LES DIFFERENTES CLASSES DE STRUCTURES DE TRAITEMENT A HAUT PARALLELISME INTRINSEQUE

3.1 - Architectures "Controlflow" et Architectures "Dataflow"

Dans l'approche classique de la synthèse de structure de traitement rapides, on peut envisager deux grandes catégories de SHPI suivant que l'on considère des dispositifs à vocation générale, et en particulier les processeurs pipe-line MISD (Multiple Instruction Stream, Single Data Stream) et MIMD (Multiple Instruction Stream, Multiple Data Stream) ou des dispositifs dont le mode d'exploitation est très spécifique et spécialisé (mais qui sont par contre très performants) et en particulier les processeurs SIMD (Single Instruction Stream, Multiple Data Stream).

Ces approches du calcul parallèle sont en fait limitées par des caractéristiques classiques telles que la commande séquentielle, les mémoires, les registres centralisés, etc... De plus elles conduisent à des contraintes quant au parallélisme et interdisent le fonctionnement asynchrone et la commande décentralisée qui sont des caractères essentiels pour une structure de calcul rapide. Ceci a donc conduit les chercheurs à introduire de nouvelles architectures ne présentant pas ces limitations, et en particulier les architectures à flux de données ("Dataflow" par opposition aux architectures plus classiques SIMD ou MIMD qui sont de type "Controlflow").

3.2 - Critères de classification des différentes catégories de structures :

Les différentes structures peuvent être classées en considérant les critères suivants :

- nature de l'architecture ("Controlflow" ou "Dataflow"),
- vitesse (de traitement sur un flux de données),
- caractère non spécifique de la structure (vocation générale),
- "efficacité" du matériel,
- volume du logiciel nécessaire (pour la programmation et la mise en oeuvre).

Nous avons pris le parti de comparer les différentes architectures de macrocellules de processeurs en mettant en évidence :

- la plus ou moins grande spécialisation de la structure (caractère "programmable" des macrocellules,
- la complexité de ces structures au plan fonctionnel (répétitivité de "motifs" élémentaires, caractère homogène ou hétérogène des macrocellules, etc...),
- la nature du systeme de communication et d'interconnexion entre macrocellules (ou entre processeurs élémentaires, cf paragraphe 4).

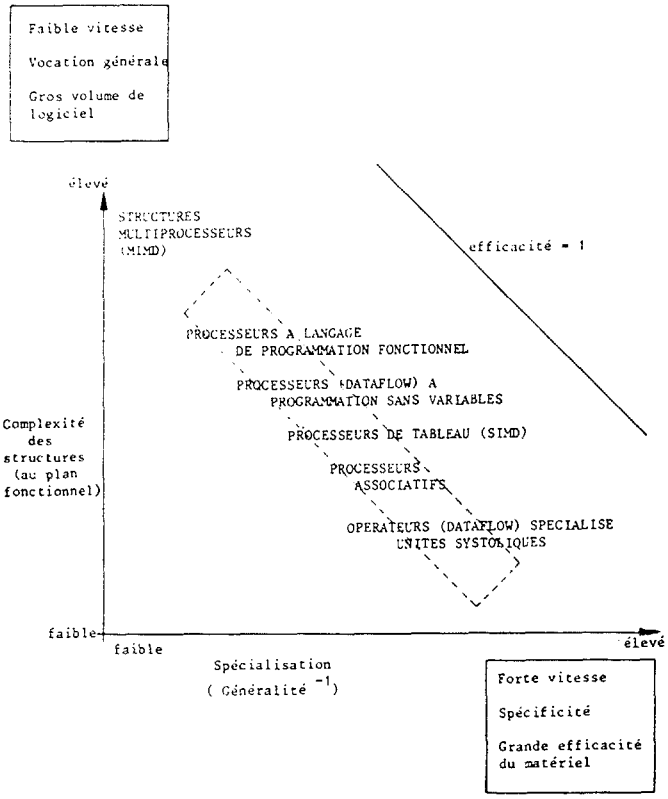

Figure 1 : Comparaison des différentes catégories de structures

3.3 - Architectures adaptées aux problèmes LCC de deuxième catégorie (traitement du signal en particulier) Approche par hiérarchisation fonctionnelle

Si l'on envisage le problème de la synthèse d'un opérateur de traitement constitué par une macrocellule (au sens du paragraphe 1.6), cette macrocellule pouvant être "dédiée" (elle permet de réaliser un algorithme déterminé), ou "non dédiée" (elle permet de réaliser l'implémentation d'une certaine classe d'algorithmes), on constate qu'il n'existe pas d'architecture interne omnivalente optimale pour tous types d'algorithmes, selon leurs formulations.

Il faut donc adopter une approche plus globale qui consiste à choisir, compte tenu d'une définition claire de la notion de macrocellule (au sens du matériel en particulier), l'implémentation la plus efficace au sens d'un certain nombre de critères (fonctionnels, matériels, logiciels, etc) d'une des formulations de l'algorithme (ou de la classe d'algorithmes) envisagé.

Ces macrocellules doivent pouvoir être associées les unes avec les autres pour former des grappes et donc des processeurs élémentaires (PE) au sens de la terminologie du paragraphe 1.6. Ces processeurs élémentaires font donc intervenir un premier niveau de structure de communication (connexion et commutation spatio-temporelle).

Un processeur élémentaire constitue une unité fonctionnelle standardisée qui est reliée à des processeurs élémentaires identiques (structure homogène) ou différents (structure hétérogène), dans le cadre d'un deuxième niveau de structure de communication.

3.4 - Architectures comportant des unités fonctionnelles multiples - opérateurs systoliques

Dans de nombreux cas, et en particulier, en traitement du signal, certaines opérations mathématiques de base forment le (ou les) noyau(x) répétitif(s) de problèmes de traitement numérique. On peut citer par exemple :

- les opérations simples sur les matrices (multiplication en particulier),
- certaines opérations d'algèbre linéaire (résolution de systèmes linéaires),
- certaines transformations discrètes (transformation de Fourier rapide),
- les opérations de filtrage linéaire, etc...

3.41 - Architectures systoliques : définition et exemples d'applications

Dans les structures systoliques (cf. figure 5) les flux de données en provenance de l'unité de mémoire et se "déplaçant" selon un ou plusieurs rythmes passent à travers de nombreux éléments de traitement avant de retourner à la mémoire (d'où l'image des systoles ou contractions du coeur, isochrones pour chacune des cavités).

Une structure systolique consiste en un ensemble de cellules de traitement interconnectées, chaque cellule étant capable d'exécuter une opération simple et les liaisons entre ces cellules étant définies de façon à minimiser la complexité des arbres ou des réseaux associés. L'information dans un système systolique s'écoule selon un ordonnancement en cascade (méthode "pipe-line") et les communications entre le système systolique et l'environnement externe se produisent uniquement sur les cellules "frontières" (sur la périphérie du système systolique).

L'assemblage des opérateurs élémentaires intervenant dans une structure systolique peut être monodimensionnel ou linéaire, ou multidimensionnel (en rectangle, en triangle, en hexagone). Un système systolique peut donc comporter des "motifs" se développant selon des directions multiples et comportant des vitesses "d'écoulement" multiples. Par opposition aux systèmes en cascade ("pipelined") classiques dans lesquels les flux sont uniquement relatifs aux résultats, dans les systèmes systoliques le flux concernant à la fois les entrées et les résultats (partiels).

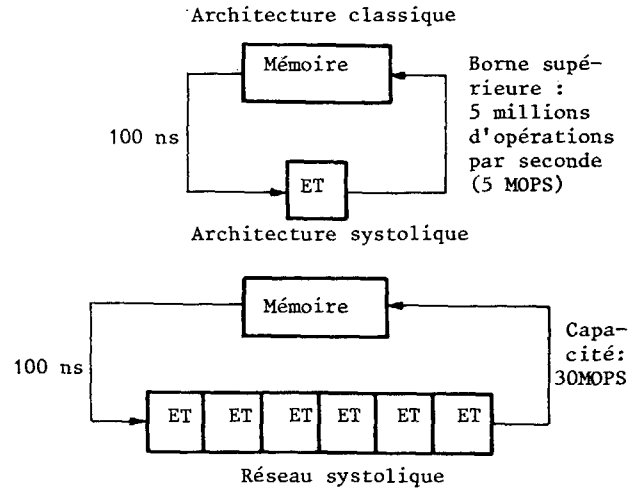

Figure 2:- Principe d'organisation d'un système systolique

ET = Opérateur Elémentaire (cellule)de traitement

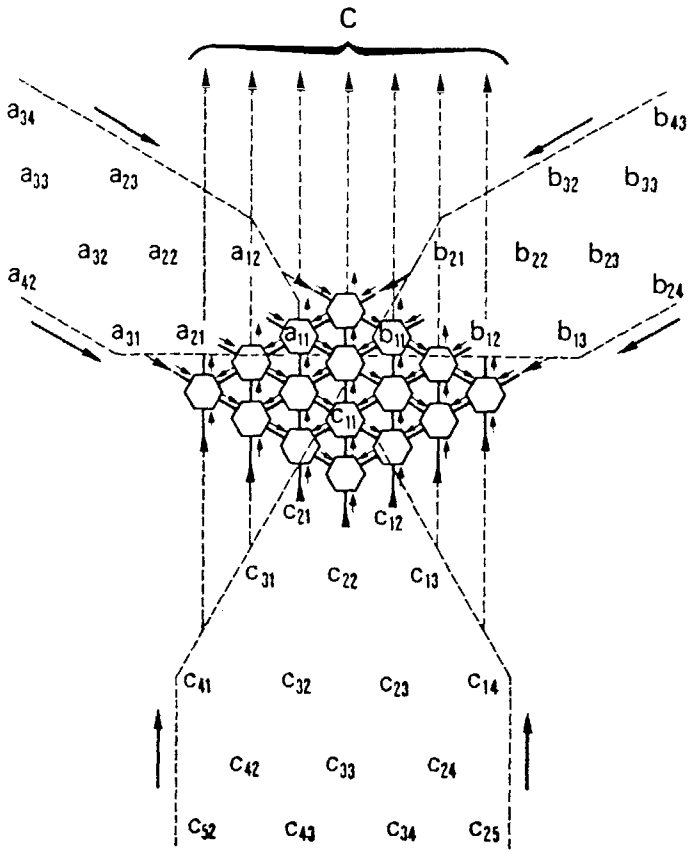

Figure 3 - Réseau systolique à connexion hexagonale pour le calcul du produit matriciel C=AxB.

3.42 - Propriétés et avantages des structures systoliques :

Les quatre principales propriétés des structures systoliques, dont découlent un certain nombre d'avantages sont les suivantes :

a) Les structures systoliques exploitent de façon intensive (multitraitement) chaque ensemble de données d'entrée.

b) Les structures systoliques exploitent des architectures à haut parallélisme intrinsèque (simultanéité des traitements).

c) Il n'existe, dans un système systolique, qu'un nombre limité de type de cellules peu complexes.

d) Les flux de données et d'instructions de contrôle ont une définition simple et sont par essence réguliers.

3.5 - Architectures programmables à flux de données : Processeurs dataflow programmables

3.51 - Modèle "Dataflow" : Définition et propriétés

Un modèle "Dataflow" ne possède aucune des deux caractéristiques du modèle de VON NEUMANN (cf. figure 4).

- Un modèle "Dataflow" ne traite que des valeurs et non les noms des positions dans lesquelles sont stockées ces valeurs (c'est-à-dire les adresses). Ce concept est également fondamental pour la synthèse de langages purement fonctionnels (cf. paragraphe 3.8 et ceci entraîne que ces langages ne sont pas élaborés et définis à partir de la notion de "mémoire globale remise à jour". Un opérateur dans ces langages produit une valeur qui est exploitée par d'autres opérateurs.

- Un modèle "Dataflow" ne possède pas de dispositif équivalent à un compteur d'instructions Une instruction est validée dès que tous ses opérandes (valeurs d'entrées nécessaires) ont été préalablement calculés et sont dipsonibles (caractère essentiellement asynchrone du modèle).

3.52 - Avantages et inconvénients d'une structure "Dataflow" :

* Les principaux avantages des structures "Dataflow", outre leurs capacités de traitement en simultanéité, résident essentiellement dans les points suivants :

- modularité et extensibilité du matériel,
- facilité de vérification du logiciel,
- limitation des erreurs commises au niveau de l'élaboration du logiciel.

Les structures de traitement basées sur ces principes présentent de nombreuses propriétés intéressantes.

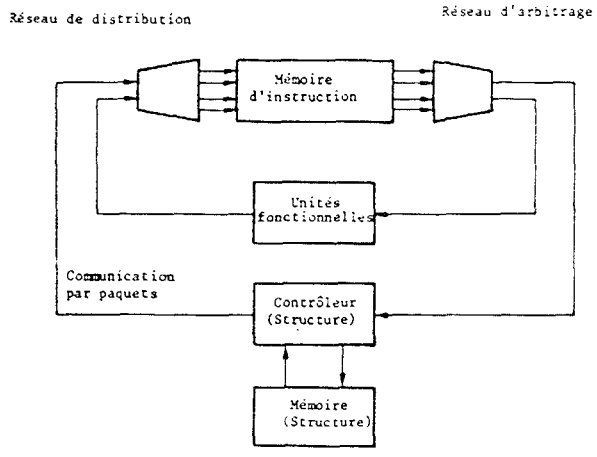

Figure 4 - Structure d'une machine "Dataflow" programmable classique

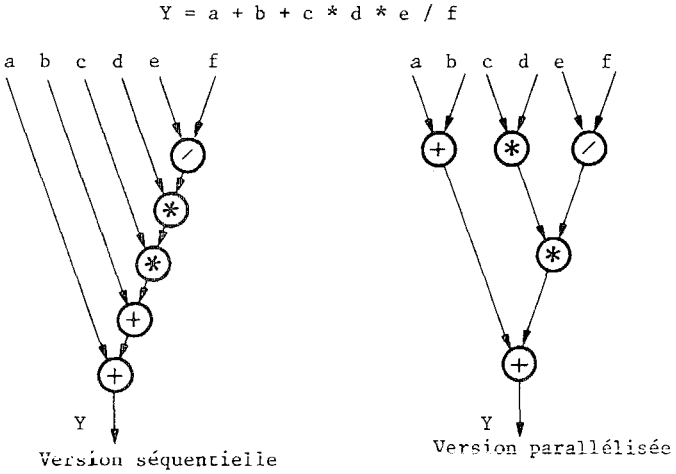

Figure 5 - Exemples de graphes "Dataflow" pour une expression simple

* Les inconvénients des structures "Dataflow" résident actuellement dans l'existence de problèmes techniques qui restent non résolus, bien qu'il ne soit pas évident que tous ces problèmes présentent tous les mêmes caractéristiques de difficulté.

3.6 - Les processeurs associatifs (Associative Processors)

On appelle mémoire associative (cf. figure 6) une mémoire organisée en parallèle par tranches de bits. Dans l'architecture correspondant à une mémoire associative, un bit d'un mot mémoire quelconque est accessible en une seule opération d'accès mémoire.
Un processeur associatif est défini à partir et autour d'une mémoire associative, en complètant les circuits d'accès aux données d'unités arithmétiques directement liées à chaque mot ou groupe de mots de la mémoire. On notera que dans un processeur associatif le problème fondamental de la manipulation des données n'est pas traité de façon optimale, bien que le domaine d'application des processeurs associatifs soit relativement limité
Il est évident que la possibilité d'accéder à des ensembles de données au moyen d'une procédure faisant intervenir la notion d'adressage par le contenu est tout particulièrement intéressante par le développement de machines optimisées pour la gestion et le traitement de bases de données.

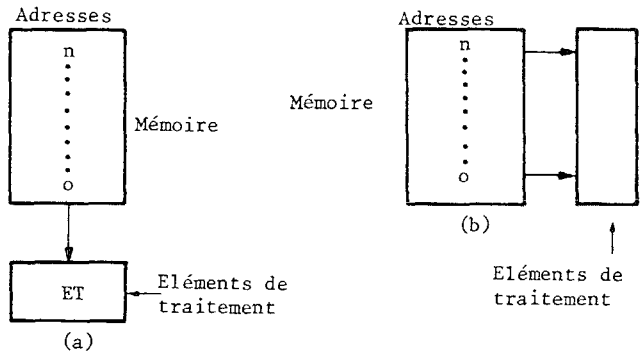

Figure 6 - Mémoires associatives - Structures parallèles par mot (a) et par tranches (b)

3.7 - Les processeurs de tableau (Array Processors)

Dans un processeur de tableau, une unité de contrôle diffuse vers un grand nombre d'éléments (ou cellules) de traitement (unités arithmétiques en général), la même instruction qui est exécutée sur des données locales par chaque cellule de traitement. Chaque unité arithmétique exécute donc la même opération sur des données différentes, les cellules de traitement pouvant unique-

ment effectuer des modifications mineures sur l'instruction qui leur est envoyée ou pouvant être programmées pour ne pas prendre en compte cette instruction.

En général les processeurs de tableau sont des machines de type SIMD (Single Instruction/Multiple Data), l'unité de contrôle étant unique et diffusant les même commandes à tous les processeurs de traitement. Les processeurs de tableau sont donc des machines dont le mode d'exploitation est très spécifique et spécialisé.

3.8 - Machines exécutant un langage de programmation fonctionnel (Functional programming Language machines)

Ces machines exécutent un langage de programmation purement fonctionnel (langage de BACKUS). Les algorithmes ayant été décrits dans une première phase en exploitent ce langage, la mise en évidence du parallélisme est automatique et immédiate, et ce sans qu'il soit nécessaire d'analyser la structure du programme et de tenter d'y détecter des parallélismes.

3.9 - Multiprocesseurs - Structures MIMD

L'existence de processeurs intégrés de faible coût et dont les fonctionnalités sont relativement complètes a considérablement accru l'intérêt porté aux structures multiprocesseurs (ou multi microprocesseurs) construites à partir de ces "modules de base" ou blocs élémentaires.

Dans ces systèmes chaque processeur est complètement programmable et peut exécuter son propre programme. Ces structures sont en général de type MIMD (Architecture "Multiple Instruction Stream, Multiple Data Stream"). Elles sont bien entendu beaucoup plus souples que les structures spécialisées, mais les opérations de contrôle associées sont beaucoup plus complexes et les dispositifs d'interconnexion correspondants présentent donc un caractère plus général (réseaux matriciels avec files d'attente aux points de commutation, etc...).

En principe les architectures multiprocesseurs doivent permettre d'atteindre un facteur d'accélération linéaire. Cependant certains facteurs empêchent la réalisation de ces performances optimales:

- Les problèmes de synchronisation
- Les problèmes associés à la nature même de l'algorithme
- Les problèmes d'"overhead"
- Les problèmes de contention
- Les problèmes d'entrée-sortie

4. - LES DISPOSITIFS DE COMMUNICATION ENTRE OPERATEURS (ET PROCESSEURS)

4.1 - Introduction

L'efficacité du système se mesure en taux de charge des différents opérateurs et processeurs de la structure considérée, la charge étant maximale quand la donnée nécessaire est disponible sur le bon processeur au bon moment.

On peut tenter de classer les différentes stratégies d'interconnexion et d'intercommunication en envisageant les trois critères suivants :

- complexité du matériel d'interconnexion rapportée au coût du réseau,
- degré de spécialisation de la réalisation développée, compte tenu de la géométrie du dispositif d'interconnexion,
- efficacité du dispositif (au sens du taux d'utilisation des processeurs).

Nous nous intéresserons plus particulièrement dans ce qui suit, aux problèmes logiques d'interconnexion (par opposition aux problèmes physiques).

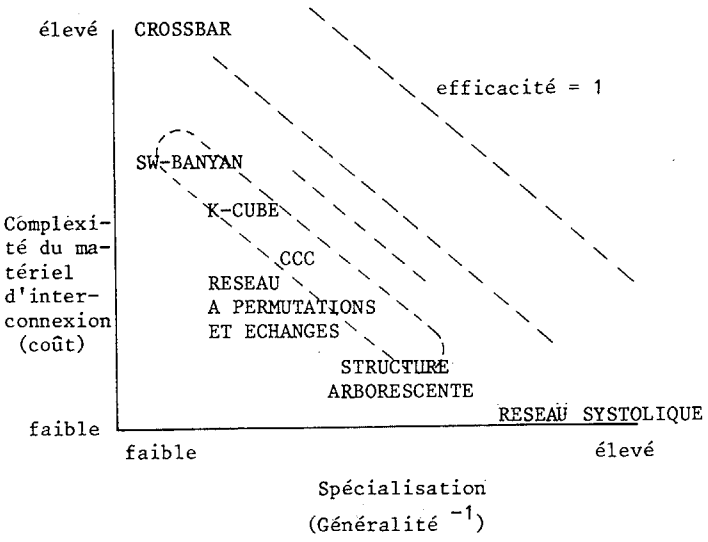

Figure 7 - Comparaison de différents dispositifs d'interconnexion

4.2 - Réseaux Crossbar

Les réseaux Crossbar ne sont pas bien adaptés, comme d'ailleurs les bus en anneau (cf. paragraphe 4.3), aux tâches d'interconnexion dans les dispositifs hautement parallèles. Dans un réseau de type Crossbar, chaque processeur est interconnecté à chaque mémoire et à chaque autre processeur. Il est donc nécessaire de disposer de n^2 commutateurs élémentaires pour interconnecter n processeurs et n mémoires. Il n'existe dans ce type de réseau aucun conflit (contention) au niveau des dispositifs de communications (mais il peut en exister pour les mémoires).

Aucun calcul d'adresse n'est à effectuer, si l'on excepte les translations d'adresse nécessaires pour établir un itinéraire au sein du réseau.

Les limitations de cette approche résident dans le nombre de dispositifs de commutation nécessaires

4.3 - Réseaux en anneau

Dans ce type de réseau, n processeurs sont connectés sur un bus en anneau.

Il est certain que cette solution est la plus simple que l'on puisse imaginer tant au niveau du matériel, qu'au niveau de l'architecture résultante. Il faut cependant noter que chacun des n processeurs ne dispose que du nième de la largeur de bande du bus.

4.4 - Réseaux systoliques et réseaux systoliques commutés

*Dans un réseau systolique, les problèmes de communication et les problèmes de traitement sont optimisés pour une classe spécifique de problèmes. Ces réseaux constituent des exemples de connexions entre processeurs selon une règle très simple qui consiste ici à n'autoriser que des liaisons d'un processeur vers ses voisins les plus proches.

*Une autre approche consiste à implémenter des réseaux systoliques commutés ou, de façon plus générale, des machines spécialisées commutées selon l'algorithme. La structure du réseau envisagé varie alors en passant d'un algorithme à un autre : on définit donc une architecture multiopérateur ou multiprocesseur qui fournit une structure d'interconnexion programmable intégrée avec les cellules (éléments) de traitement. Cette structure d'interconnexion est construite autour d'un treillis de dispositifs de communication, d'un ensemble de processeurs, et d'un contrôleur.

*On peut également envisager d'autres approches comme celles des graphes partiellement ordonnés divisés en niveaux dinstincts (BANYAN NETWORK). Chaque noeud (correspondant à un commutateur) comporte un certain nombre d'arêtes orientées vers lui et un certain nombre d'arêtes s'étalant à partir de lui dans la direction des noeuds dans le niveau suivant.

4.5 - Interconnexion faisant intervenir des architectures cubiques

a) Le k-Cube

Le k-cube binaire permet d'interconnecter $n = 2^k$ processeurs ou opérateurs. Si l'on considère que les processeurs sont situés aux sommets d'un cube dans un espace à k dimensions, les arcs intervenant dans les connexions sont les arêtes du cube.

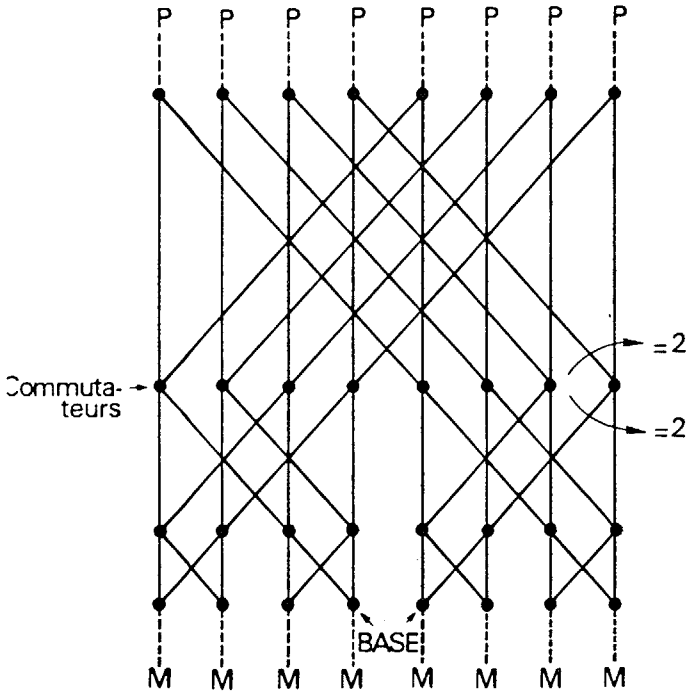

Figure 8 - Réseau BANYAN.

b) Réseaux en anneau répartis sur une structure cubique : CCC et réseaux à permutation :

Les opérateurs peuvent être organisés en cascade linéaire ayant des structures d'anneaux (structures série), chaque groupe linéaire étant situé sur l'un des sommets d'un hypercube (CCC ou Cube-Connected Cycles - cf. figure 11).

*Cette organisation, qui présente l'avantage de conduire à un nombre de connexions constant par processeur ou opérateur, peut être considérée comme une implémentation ou une émulation de la structure k-cubique. Elle présente des avantages marquants dans le cas où le problème considéré peut être décomposé de façon récursive en problèmes de taille plus faible, les solutions de ces problèmes partiels étant associés de façon à déterminer la solution du problème global.

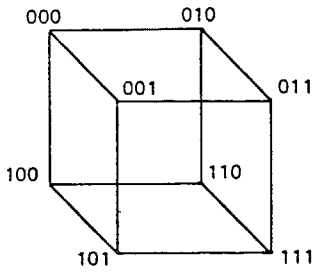

Figure 9 : Cube pour k = 3

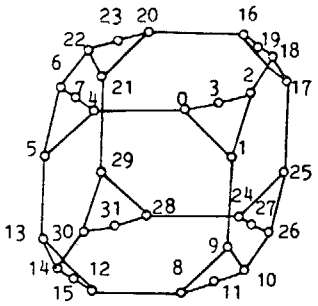

Figure 10 - Réseau en anneaux répartis sur une structure cubique (CCC)

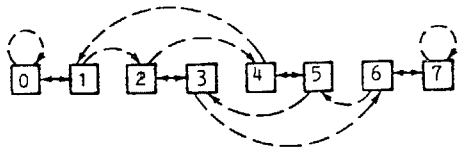

Figure 11 - Réseau à échanges (lignes continues) et permutations (tirets)

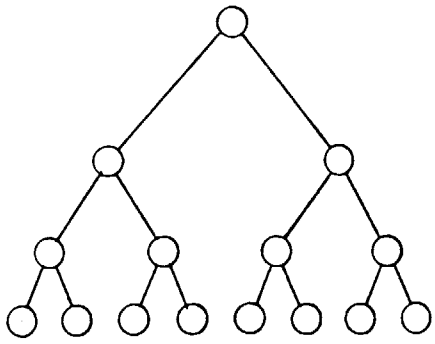

Figure 12 - Structure arborescente binaire

4.6 - Structures arborescentes

Sur la figure 12 on représente une structure arborescente binaire, chaque processeur ou opérateur étant connecté à ses deux "enfants" (cas de la structure binaire) et à son "père" unique.

Les machines à structure arborescente sont particulièrement bien adaptées aux algorithmes ayant eux-mêmes une structure arborescente, et par exemple aux problèmes de tri, de multiplication de matrices. ..