

# Architectures parallèles spécialisées pour le traitement d'image

François Charot

► **To cite this version:**

François Charot. Architectures parallèles spécialisées pour le traitement d'image. [Rapport de recherche] RR-1978, INRIA. 1993. <inria-00074694>

**HAL Id: inria-00074694**

**<https://hal.inria.fr/inria-00074694>**

Submitted on 24 May 2006

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

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

*Architectures parallèles  
spécialisées pour le traitement d'image*

François Charot

**N° 1978**

Avril 1993

PROGRAMME 1

Architectures parallèles,  
bases de données,  
réseaux et systèmes distribués



*Rapport  
de recherche*

1993





## Architectures parallèles spécialisées pour le traitement d'image

François Charot\*

Programme 1 — Architectures parallèles, bases de données, réseaux  
et systèmes distribués  
Projet API

Rapport de recherche n°1978 — Avril 1993 — 42 pages

**Résumé :** Un domaine d'applications comme le traitement d'image requiert une telle puissance de calcul que seules les architectures parallèles spécialisées "taillées sur mesure" peuvent satisfaire les contraintes de vitesse, de performance et d'encombrement.

La réalisation d'architectures parallèles spécialisées pour l'image est abordée de différentes façons. Dans la catégorie des machines SIMD, on note les réseaux bidimensionnels à base de processeurs bit-série (MPP, GAPP, DAP), les réseaux linéaires réalisés à partir de processeurs plus performants (16 voire 32 bits), (CLIP7, SLAP, SYMPATI, PRINCETON). Parmi les machines MIMD, on note essentiellement les machines WARP, iWARP, et les approches à base de processeurs de traitement de signal (DSP) supportant ainsi différents types de parallélisme (systolique, pipeline, ...). Aussi, les approches employées à ce jour sont variées et leur comparaison difficile.

L'objectif de cet article est de proposer un panorama des architectures parallèles spécialisées pour l'image jugées les plus significatives. Pour chacune des réalisations décrites, l'accent est mis sur l'aspect qui présente le plus d'intérêt (structure du processeur, organisation de la machine, mode de fonctionnement, système d'interconnexion).

**Mots-clé :** Architecture pour le traitement d'image, SIMD, MIMD

*(Abstract: pto)*

\*charot@irisa.fr

# Specialized parallel architectures for image processing

**Abstract:** A field of application such as image processing demands so much computational power that only specialized “tailor made” parallel architectures are capable of satisfying the constraints of speed, performance and capacity.

The implementation of parallel architectures which are specialized for processing images is approached in different ways. For example, in the class of SIMD machines, there are two-dimensional arrays made up of bit-serial processing elements (MPP, GAPP, DAP), and linear arrays made up of more powerful processors (16 or 32-bit), (CLIP7, SLAP, SYMPATI, PRINCETON). In the class of MIMD machines, there have been both the WARP and iWARP machines and the DSP-based approaches, each based on different styles of parallelism (pipeline and/or systolic). Since there are a variety of approaches to parallelism in these architectures, their comparison is difficult.

The goal of this article is to offer an overview of the most important image processing architectures. For each implementation described, we emphasize the most significant architectural feature (processor architecture, machine organization, operation, network topology).

**Key-words:** Architecture for image processing, SIMD, MIMD

## **Introduction**

Le traitement numérique d'image est un domaine particulièrement vaste, les applications sont nombreuses et variées : gestion de bases de données images, compression d'images fixes ou animées en vue de leur stockage ou leur transmission, analyse et compréhension d'images, etc. Toute manipulation d'images met en jeu un volume considérable de données ; celui-ci est lié à la dimension des images manipulées par l'application et croît lorsque, pour fournir un élément d'interprétation, il s'agit d'avoir accès à plusieurs images consécutives d'une séquence.

Généralement, chaque image passe par plusieurs niveaux de traitements, des traitements numériques de bas niveau (niveau du pixel) aux traitements symboliques de haut niveau (niveau primitive). A chacun de ces niveaux et plus particulièrement au niveau le plus bas, l'algorithme est très dispendieux en calculs. Aussi, le volume de calcul inhérent à de nombreuses applications de traitement d'image excède la capacité de traitement de toute machine séquentielle. Le problème est encore plus critique dans un contexte temps réel ou une séquence d'images doit être traitée dans un temps très court.

La structure de données couramment manipulée en traitement d'image est le tableau bidimensionnel qui représente la distribution spatiale des valeurs de luminosité (niveaux de gris ou couleur). Dans de nombreux algorithmes, le traitement à réaliser consiste en l'application d'un ensemble d'opérations en chaque point de la structure de données image. Aussi ces opérations peuvent être réalisées en parallèle pour chaque élément du tableau.

Les systèmes matériels utilisés s'étendent de la simple carte PC aux énormes supercalculateurs et tirent parti d'une vaste gamme de composants allant du processeur classique aux circuits intégrés VLSI les plus spécifiques. En conséquence, les machines parallèles couramment employées en image sont aussi bien des architectures à usage général que des architectures dédiées. Les architectures parallèles à usage général sont intéressantes pour leur flexibilité ; elles supportent une large gamme d'algorithmes avec des performances somme toute intéressantes et permettent ainsi une validation des spécifications de l'application considérée. Quant aux architectures spécialisées ou dédiées, elles permettent une validation fine de l'application de traitement d'image ; elles offrent des performances supérieures en terme de vitesse de traitement, comme par exemple le temps réel et disposent de mécanismes d'entrée/sortie vidéo.

Plusieurs raisons militent en faveur d'une approche "architecture parallèle spécialisée" où le système est "taillé sur mesure" pour une classe de problèmes ;

ce sont les caractéristiques du problème qui déterminent la structure et le contrôle du système. Les principales raisons sont les suivantes.

- Il est souvent nécessaire d'atteindre la plus grande vitesse de traitement possible (systèmes temps réel).
- Le système doit très souvent être compact (matériel embarqué), Il est dans ce cas primordial d'optimiser l'architecture pour une classe de problèmes et de minimiser ainsi sa taille.
- Les progrès de la technologie des circuits intégrés VLSI ont rendu possible la réalisation de systèmes ayant un grand nombre de processeurs, chaque processeur intègre des fonctionnalités de traitement minimales ou réalise une fonction particulière. Il est ainsi plus facile de dimensionner le système.

Une présentation exhaustive de toutes les réalisations de machines pour le traitement d'image est bien entendu impossible. L'objectif visé par cet article est proposer un panorama des architectures parallèles spécialisées jugées les plus significatives. Pour chacune des réalisations décrites, l'accent est mis sur l'aspect qui présente le plus d'intérêt (structure du processeur, mode de fonctionnement, système d'interconnexion, etc).

La première partie de l'article présente les deux principes de traitement sur lesquels repose la plupart des mises en œuvre en traitement d'image : le traitement pipeline et le parallélisme de données. Une classification des architectures spécialisées pour l'image est ensuite proposée. Les parties deux et trois concernent les approches SIMD, des réalisations à base de réseaux bidimensionnels et linéaires sont successivement décrites. La quatrième partie est consacrée aux approches pipelines. Les structures pyramidales sont évoquées dans la cinquième partie. Les architectures MIMD sont considérées dans la sixième partie. Enfin, dans une dernière partie, une comparaison de ces différentes approches est proposée.

## 1 Parallélisme et image

Les architectures parallèles employées en traitement d'image reposent principalement sur deux principes de traitement.

- Le traitement pipeline. Une suite d'opérations élémentaires est appliquée à l'image. La technique du pipeline est une méthode simple qui consiste à

transmettre l'image de façon séquentielle à travers une suite d'opérateurs optimisés. Les opérateurs utilisés peuvent être des opérateurs ponctuels (opérations point à point comme la transformation de pixels, la moyenne et différence sur plusieurs images), locaux (convolution, filtrage, opérations morphologiques, dilatation, érosion) ou globaux (histogramme). L'emploi de ces opérateurs conduit à des architectures simples, efficaces mais très spécialisées et peu flexibles puisque les chemins de données entre opérateurs sont câblés.

- Le parallélisme de données. Les unités opératives sont dupliquées : il y a autant de processeurs que de pixels dans l'image. Des traitements simples et identiques sont appliqués en parallèle sur l'ensemble des pixels de l'image. Cette approche conduit assez naturellement à des architectures massivement parallèles, réalisées à partir d'éléments de traitement simples, et plutôt spécialisés. C'est à ce jour, l'approche retenue dans bon nombre de travaux d'architectures parallèles pour le traitement d'image.

Ces deux principes de traitement ne suffisent pas pour proposer une classification des machines de traitement d'image qui soit intéressante et représentative des machines couramment employées. Pour pouvoir établir une réelle comparaison des architectures, d'autres critères sont envisageables : la topologie du réseau d'interconnexion qui relie les processeurs, l'arithmétique supportée, le mode de fonctionnement (synchrone, asynchrone), le spectre des applications considérées, ou encore la taille du système, son coût, ses performances, sa polyvalence et ses facilités de mise en œuvre.

Aussi avant de proposer une classification des machines de traitement d'image, nous discutons quelques uns des critères qui, à notre avis, permettent de préciser la classification.

## 1.1 Classification des architectures parallèles

Les architectures parallèles sont le plus souvent classées selon leur mode de fonctionnement, c'est à dire la multiplicité des flots de contrôle et de données (SIMD, pipeline, MIMD). Néanmoins, ce critère de fonctionnement conduit à une classification très grossière des machines et s'avère être inadapté pour couvrir le très large éventail des implémentations existantes.



Nous discutons ici les critères qui permettent de préciser les styles de machines de traitement d'image [MaLL88] : la topologie du réseau d'interconnexion, l'autonomie des processeurs, leur structure interne, l'arithmétique supportée.

La topologie du réseau d'interconnexion qui relie les processeurs entre eux est un critère important. En effet, le réseau d'interconnexion est un point particulièrement critique dans la réalisation d'une machine parallèle. Nous constatons que les topologies des architectures parallèles pour l'image sont variées (grille, tore, anneau, réseau linéaire, pyramide, hypercube).

Le second point important concerne l'autonomie des processeurs. Compte tenu de la nature de l'image, les architectures SIMD ont toujours constitué un centre d'intérêt important. Un grand nombre d'architectures pour l'image supportent ce mode de fonctionnement. L'intérêt pour le parallélisme massif et l'évolution de la technologie des circuits intégrés y ont très largement contribué. Une machine SIMD consiste en une unité de contrôle unique et un ensemble d'éléments de traitement (processeurs). L'unité de contrôle a la charge de diffuser l'instruction qui est simultanément exécutée par tous les processeurs. Une des principales limitations du fonctionnement SIMD résulte du comportement identique de tous les processeurs qui exécutent tous cette même instruction : l'autonomie du processeur constitue une extension de ce modèle de fonctionnement.

Trois niveaux d'autonomie peuvent être injectés dans une architecture SIMD de base.

- L'autonomie d'opération permet à des opérations différentes de s'exécuter simultanément, ce ce afin d'accroître le parallélisme.
- L'autonomie d'adressage permet à chaque processeur d'accéder à des adresses mémoire différentes.
- L'autonomie de connexion permet de modifier dynamiquement la topologie du réseau d'interconnexion selon des conditions locales à chaque processeur.

L'organisation de l'architecture du processeur élémentaire constitue également un critère de comparaison intéressant. L'examen détaillé d'un certain nombre d'architectures montre la diversité des styles de processeurs (du processeur bit-série organisé autour d'une UAL et de quelques registres 1 bit aux processeurs disposant de plusieurs unités fonctionnelles 16 voire 32 bits travaillant en parallèle). Aussi le grain de parallélisme est-il un autre paramètre important. Les données couramment manipulées en image sont aussi bien des bits que des mots codés sur

8, 16 voire 32 bits. Néanmoins, il faut souligner, qu'à ce jour, les architectures massivement parallèles spécialisées pour l'image ont très souvent privilégiées le traitement bit-série.

### 1.1.1 Autonomie d'opération

L'autonomie d'opération vise à permettre l'exécution simultanée d'opérations différentes par les processeurs. Cela ne veut pas nécessairement dire que chaque processeur exécute un programme différent (mode de contrôle MIMD) mais plutôt que les processeurs n'exécutent pas tous la même opération comme en mode SIMD.

Différents types d'autonomie d'opération existent.

- La notion de processeurs actifs (exécutant l'instruction) et de processeurs inactifs (n'exécutant pas l'instruction) existe dans la plupart des machines SIMD. Ce mécanisme est supporté via un registre de masquage ou d'activité qui inhibe ou pas les chargements mémoire ou registre.
- Dans un système comme CLIP7 [FoMD88], l'instruction diffusée par l'unité de contrôle est interprétée différemment dans chacun des processeurs. Un registre interne au processeur est vu comme une extension du code opération de l'UAL.
- La notion de groupes de processeurs dans les approches multi-SIMD offre également une autonomie d'opération puisque chaque groupe de processeurs a son propre contrôleur.

### 1.1.2 Autonomie d'adressage

L'autonomie d'adressage permet à chaque processeur de la machine SIMD de calculer son adresse d'accès à la mémoire localement ou de modifier l'adresse diffusée par le contrôleur. Cette autonomie d'adressage facilite la mise en œuvre de tables de look-up et permet l'adressage indirect.

### 1.1.3 Autonomie de connexion

L'autonomie de connexion consiste à adjoindre à chaque processeur un mécanisme matériel capable de réaliser les interconnexions entre les liens physiques sur la base des spécifications de l'instruction. Cette autonomie peut être appliquée à toute

topologie ; cependant la structure torique est celle qui se prête le mieux à cette extension.

## 1.2 Familles de machines

Comme discuté dans le paragraphe précédent, toute classification dépend fortement des critères retenus (mode de fonctionnement, structure du réseau d'interconnexion, etc). Aussi la répartition proposée ci-après est à la fois basée sur le style de parallélisme mis en œuvre (parallélisme de type pipeline et/ou parallélisme de type géométrique caractérisé par une répartition de l'image entre les processeurs), sur le mode de fonctionnement (SIMD, MIMD, pipeline) et la topologie du réseau. Les familles suivantes, illustrées par la figure 1, peuvent être identifiées.

1. Les systèmes reposant sur une interconnexion en maille opèrent sur la totalité de l'image et réalisent des opérations locales sur tous les pixels de l'image simultanément (CLIP4 [Duff78], MPP [Batc80], DAP [OIRe85], GAPP [Kung88]). Il s'agit de réseaux bidimensionnels de processeurs opérant en mode SIMD. Ces architectures mettent en œuvre un parallélisme géométrique. On peut considérer les réseaux linéaires de processeurs comme appartenant à cette classe de machines. Ces machines reposent sur un balayage ligne de l'image (CLIP7 [FoMD88], SLAP [Fish86], AIS-5000 [ScWi88], PRINCETON [Chin88], SYMPATI [Adam89] [JBEL88]).
2. Les systèmes pipelines tels les différents types de "Cytocomputer" et les réseaux systoliques de processeurs. Ces systèmes réalisent des séquences d'opérations sur un flot de données d'entrée. Dès qu'une opération a été effectuée sur une partie des données, l'opération suivante peut être amorcée sur cette partie de données. Ces architectures balayent l'image et traitent un petit voisinage à chaque étape. La plupart des systèmes de vision industriels reposent sur cette approche (Cytocomputer [LoMc80], PIPE [KeSL85]). Ces architectures mettent en œuvre un parallélisme de type pipeline.
3. Les systèmes pyramidaux permettent de réaliser des opérations globales par des techniques "diviser pour régner" en un nombre d'étapes de traitement qui croît de façon logarithmique en fonction de la taille des images (SPHINX [MeND91], PCLIP PAPIA GAM [Uhr87]). Ce style d'architectures trouve sa raison d'être dans les traitements multi-échelles (ondelettes, pyramides gaussiennes) ou des traitements récursifs. Par traitement multi-échelle, on entend

des méthodes qui appréhendent l'image à différents niveaux de résolution. Le fonctionnement est de type multi-SIMD.

4. Des architectures dont l'organisation n'est pas directement calquée sur la structure de l'image ou le style de calcul à réaliser (PASM [Sieg81], WARP [Anna87], machine à base de processeurs de traitement de signal DSP, [Head88], [Tami89], etc) sont également couramment utilisées en traitement d'image. Ces architectures, MIMD, permettent la mise en œuvre de plusieurs styles de parallélisme.
5. Les hypercubes qui combinent les avantages des structures pyramidales et des réseaux maillés (iPSC, Connection Machine [Hill85]). Ces architectures à usage général ne seront pas décrites ici.

## 2 Architectures SIMD à réseau maillé 2D

De nombreuses opérations de traitement d'images sont caractérisées par un traitement local dû à la nature stationnaire de la donnée image. L'approche qui consiste à découper l'image en régions de même taille et à répartir les régions entre des processeurs identiques traitant chacun leur propre région est largement employée.

Ces architectures opèrent sur la totalité de l'image en parallèle. La dimension du réseau ( $n \times n$  processeurs) peut être inférieure à la taille de l'image à traiter ( $m \times m$  pixels). Dans ce cas, l'image est répartie entre les différents processeurs selon l'une des deux approches suivantes.

1. L'approche la plus couramment utilisée consiste à distribuer l'image entre tous les processeurs en mémorisant chaque pixel  $(i, j)$  d'une sous-image de taille  $n \times n$  dans le même module mémoire, l'image est donc préalablement divisée en sous-images de taille  $n \times n$ , comme illustré par la figure 2.
2. La seconde approche consiste à affecter à chaque processeur une sous-image de taille  $m/n \times m/n$  (figure 3).

Les réalisations qui mettent en œuvre ce style de parallélisme sont nombreuses. Les systèmes les plus représentatifs de cette classe sont CLIP4, DAP, GAPP, MPP, YUPPIE, etc. Les caractéristiques de ces architectures sont les suivantes.

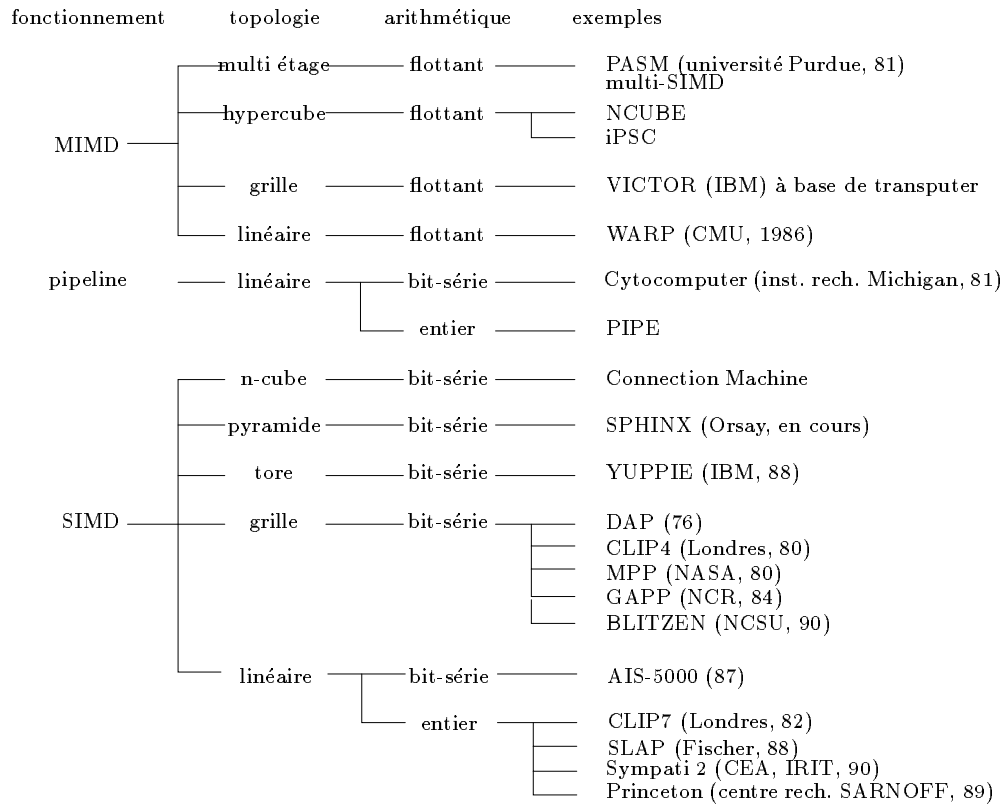


Figure 1 : Classification des machines de traitement d'images.

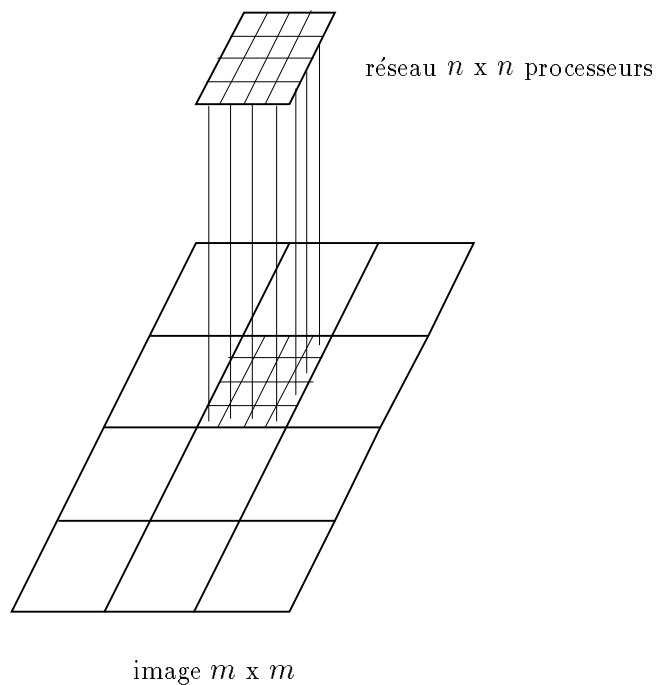


Figure 2 : Division de l'image en blocs  $n \times n$ .

- Les processeurs élémentaires sont de type bit-série. Le processeur intègre une UAL, quelques registres et de la logique spécifique à la mise en œuvre de communications bidirectionnelles avec ses proches voisins.
- Le mode de fonctionnement est SIMD c'est à dire que tous les processeurs exécutent simultanément la même instruction.

## 2.1 CLIP4 (université de Londres)

La machine CLIP4 [Duff78] (réalisée à l'université de Londres par Duff, 1980) est conçue autour d'une matrice  $96 \times 96$  de processeurs booléens bit-série. Chaque processeur a une mémoire de 32 mots de 1 bit et est relié à ces huit voisins. Une fenêtre de données correspondant à une trame de  $96 \times 96 \times 6$  bits peut être

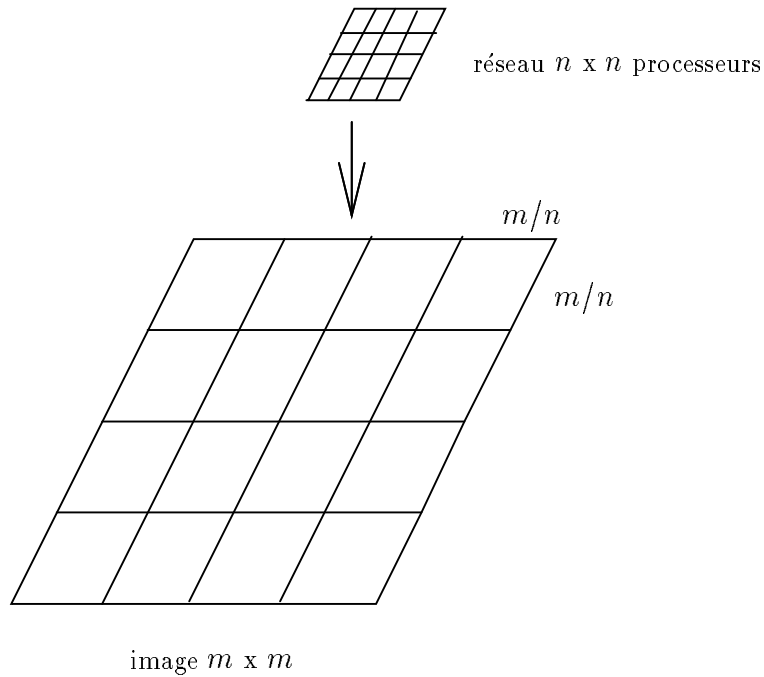


Figure 3 : Division de l'image en blocs  $m/n \times m/n$

mémorisée dans 6 registres à décalage (1 par plan de bit), un tel plan de bit est converti en une matrice binaire  $96 \times 96$  pouvant ainsi être transférée dans la matrice de processeurs. CLIP est très performant pour les opérations logiques portant sur une fenêtre binaire  $3 \times 3$  (1 cycle), par contre des opérations de convolution  $3 \times 3$  exigent 2000 cycles.

## 2.2 MPP (Goodyear)

La machine MPP (réalisée pour la NASA à Goodyear par Batcher, 1980) correspond à un réseau de 16 384 processeurs (matrice  $128 \times 128$  processeurs).

Chaque processeur (figure 4), bit-série, est plus orienté calcul arithmétique que CLIP4. Il a une mémoire de 1K bit et est relié à ses quatre plus proches voisins (N, S, E, O).

Les données entrent et sortent dans le système par l'intermédiaire de deux registres (*staging memory*) de 2,5 millions d'octets. Dans ces registres, le flux de données est converti en plans de bits de 16 384 bits, pouvant être introduits dans le réseau à la vitesse de 160 millions d'octets par seconde. Les performances sont de l'ordre de 6500 millions d'additions par seconde sur des entiers 8 bits (cycle de base 100 ns).

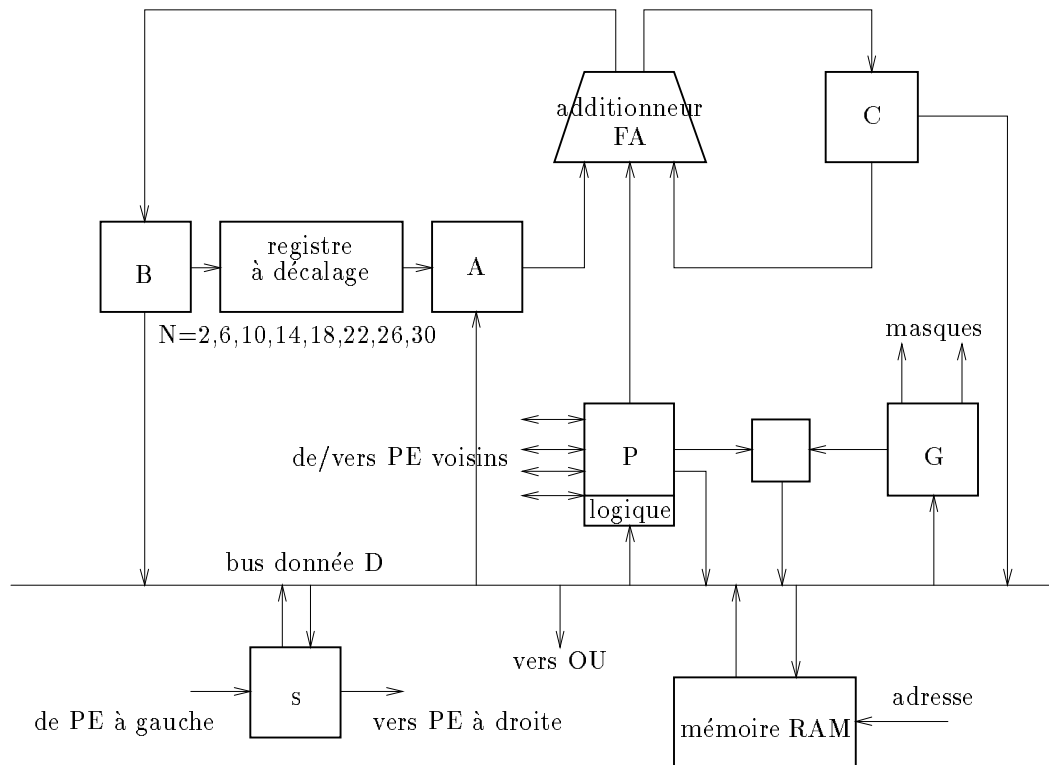


Figure 4 : Organisation générale du processeur MPP

La machine BLITZEN, (projet du triangle de recherche de la Caroline du nord, 1989), [BDHR90] est une évolution de la machine MPP. Les apports essentiels concernent l'intégration de la mémoire de données sur le circuit, une autonomie d'adressage (modification locale de l'adressage mémoire), la possibilité d'opéra-



tions arithmétiques conditionnelles, l'interconnexion avec huit voisins. Le circuit intègre 128 processeurs (tableau 8 x 16) et comprend plus d'un million de transistors, chaque processeur dispose d'une mémoire locale de 1K bits.

### 2.3 Tore polymorphique (IBM, université de Gênes)

Le Tore polymorphique (1987) [LiMa89] constitue une étude intéressante de réseau bidimensionnel configurable. L'architecture consiste en un réseau physique (PNET) comportant à chaque nœud un réseau interne programmable (INET). Une architecture à  $n \times n$  processeurs est un grille  $n \times n$  dont les extrémités sont reliées en tore ou en spirale. Un processeur  $P_{i,j}$  situé à chaque jonction de la grille, dispose de 4 ports (ou liens physiques)  $(N_{i,j}, S_{i,j}, E_{i,j}, O_{i,j})$ . Ces 4 ports sont à la fois reliés aux réseaux PNET et INET et constituent l'interface entre les deux. Le réseau PNET à l'exception de la sélection du mode tore ou spirale est un réseau dont les connexions sont fixées et non programmables.

Le réseau INET dont l'organisation est illustrée par la figure 5 est totalement programmable. Un commutateur  $INET_{i,j}$  à un nœud du réseau, est un graphe complet de 4 ports  $(N_{i,j}, S_{i,j}, E_{i,j}, O_{i,j})$  c'est à dire que chaque port peut être relié à l'un quelconque des autres ports.

Chaque INET peut être programmé différemment. La connexion INET est représentée par la fonction :  $SHORTPORT(pidcondition)\{g1\}, \{g2\}$  ou  $pid$  est l'identification du processeur et indique que la fonction est locale et dépendante de la position de INET. Les paramètres  $g1$  et  $g2$  sont des groupes de ports connectés entre eux. Par exemple,  $\{E, O\}$  signifie que  $E$  et  $O$  sont dans le même groupe, trois ports peuvent être dans le même groupe  $\{N, E, O\}$ . L'effet de la fonction est de rendre équivalent les niveaux logiques des ports concernés. La fonction  $SHORTPORT\{E, O\}$  assure que les signaux sur les ports  $E$  et  $O$  sont au même niveau logique. Cette fonction fournit un mécanisme de communication entre processeurs distants du réseau PNET.

Ce concept de configurabilité du réseau de communication peut être aussi bien appliqué à des processeurs bit-série qu'à une approche parallèle.

#### Implémentation

Le coût inhérent à cet ajout de fonctionnalité a été évalué au travers d'une implémentation VLSI réalisée à Yorktown : le système YUPPIE [MaLi89]. Ce coût concerne :

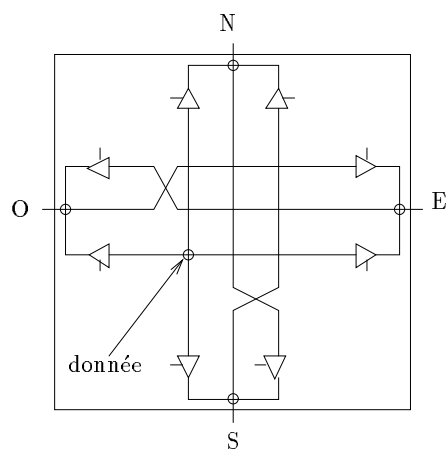


Figure 5 : Le réseau interne INET de la machine Tore.

- le surcoût en silicium lié au fait d'avoir un réseau INET par processeur,
- le délai de propagation introduit par l'INET.

Concernant le délai de propagation, la stratégie retenue consiste à rendre le temps de cycle de la machine dépendant de la longueur du chemin à parcourir entre les nœuds qui échangent des données (implémenté par un générateur d'horloge programmable commandé par une table dont le contenu indique le temps de cycle correspondant à la distance de communication).

Le circuit intègre 16 nœuds, organisés en grille 4 x 4. Chaque nœud consiste en l'unité de calcul, un INET, et une mémoire de données locale de 256 registres 1 bit. Le circuit fabriqué en technologie CMOS  $2\mu\text{m}$  comprend 50 000 transistors (67 broches) pour une taille de  $32\text{ mm}^2$  (24% pour la mémoire locale, 38% pour les 16 unités de calcul et 12% pour les 16 INETs). L'unité de calcul comporte une UAL 1 bit, et 5 registres (carry, 2 registres de données, 2 registres de contrôle).

### 3 Architectures SIMD à structure ligne

Les réseaux linéaires sont un cas particulier des réseaux bidimensionnels : ceux-ci mettent en œuvre un parallélisme géométrique, l'image est découpée en régions. Le nombre de régions est généralement égal au nombre de processeurs de la machine, la région pouvant être réduite à une ligne ou une colonne de l'image.

Les réalisations les plus représentatives sont décrites dans ce paragraphe. Leurs principales caractéristiques sont les suivantes :

- Le mode de fonctionnement est SIMD c'est à dire que tous les processeurs exécutent simultanément la même instruction.
- La souplesse de dimensionnement autorise l'utilisation de réseaux dont la dimension est égale à la taille de l'image (image  $n \times n$ , réseau à  $n$  processeurs).
- Les processeurs élémentaires disposent de davantage de fonctionnalités que ceux des structures bidimensionnelles. En effet, la plupart (à l'exception de AIS-5000) traitent des entiers (8 à 16 bits selon les réalisations).

#### 3.1 Machine CLIP7 (Université de Londres)

Le système CLIP7 [FoMD88] conçu en 1982, résulte d'une série d'améliorations de la machine CLIP4.

Les améliorations suivantes ont été apportées à l'architecture CLIP4 :

- mémoire locale processeur plus importante,
- recouvrement des entrées/sorties et des calculs,
- contrôle local au processeur plus important.

Le circuit CLIP7 (16 bits, technologie CMOS  $5 \mu\text{m}$ , 5 Mhz, 6800 transistors,  $20\text{mm}^2$ , 64 broches) contient une UAL (16 opérations logiques, addition, soustraction), une unité de décalages, 4 registres temporaires, un port mémoire externe 8 bits, des registres d'entrée/sortie pour les échanges série entre processeurs (figure 6).

Un système à base de CLIP7 a été développé et consiste en un réseau linéaire de 256 processeurs pour le traitement d'images de taille  $256 \times 256$  pixels. Le

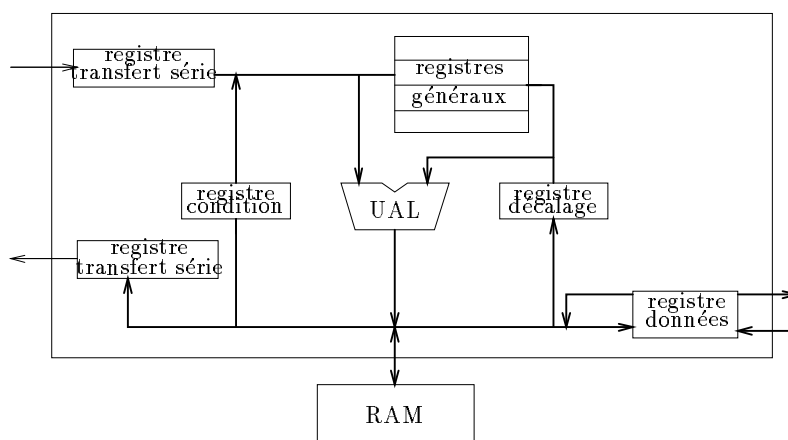


Figure 6 : Les chemins de données du circuit CLIP7.

processeur élémentaire du réseau est construit autour de 2 circuits CLIP7 : le premier assure la manipulation des données, les échanges avec les processeurs voisins, le second est vu comme coprocesseur et assure les fonctions d'accès à la mémoire de données externe, la génération d'adresse (figure 7).

Les réseaux bidimensionnels comme CLIP4 et MPP n'incluent que très peu d'autonomie. Le système à base de CLIP7 permet à chaque élément du réseau une autonomie d'adressage complète et une certaine flexibilité pour sélectionner la fonction à exécuter en chaque point.

### 3.2 Machine SLAP (Scan Line Array Processor) (Université de Carnegie-Mellon)

L'architecture développée par Fisher à CMU [Fish86] [FiHR88] repose sur une approche à balayage ligne : à un instant donné une ligne complète de l'image est traitée en parallèle, chaque processeur opère sur un pixel de la ligne (adaptation de la taille du réseau au nombre de colonnes de l'image). En conséquence, un processeur traite une colonne complète de l'image. Les processeurs sont localement

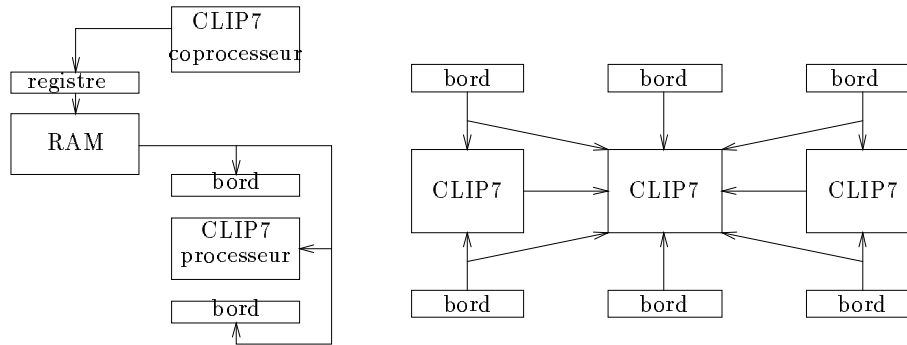


Figure 7 : Système à base de circuits CLIP7.

connectés (réseau linéaire) et peuvent échanger des données à chaque cycle par l'intermédiaire d'un lien bidirectionnel (figure 9).

Le processeur élémentaire (PE) [FiHR87] (16 bits) comporte deux unités de calcul en arithmétique entière 16 bits (UAL<sup>1</sup>, unité de décalage/rotation), une unité de 32 registres double port supportant une lecture et une écriture à chaque cycle, un registre de communication (appelé registre de voisinage) avec les PE voisins. Chaque PE (figure 8) intègre un registre à décalage vidéo 8 bits. En mode temps-réel, une image est transmise au rythme vidéo par l'intermédiaire du registre à décalage, lorsqu'une ligne d'image a été intégralement transmise, elle peut être traitée. Dans le même temps la ligne suivante est reçue. La mémoire locale est externe au processeur, le processeur n'a pas d'interface mémoire particulière. Le registre vidéo sert de registre de données, le registre de voisinage est utilisé pour les adresses.

Un système prototype à 512 processeurs (4 processeurs par circuits, soit environ 45000 transistors, 10000 transistors par processeur) est en cours de développement. Un temps de cycle de 125ns permettrait l'exécution en temps de réel de 500 instructions pour chaque pixel d'une image 512 x 512. Une carte au format SUN peut contenir jusqu'à 64 circuits auquel il faut ajouter la mémoire locale à chaque processeur (4K octets par processeur). Un système à 512 processeurs nécessite 3 cartes de ce type (2 cartes processeur, et une carte contrôleur).

<sup>1</sup>Une multiplication nécessite dix instructions.

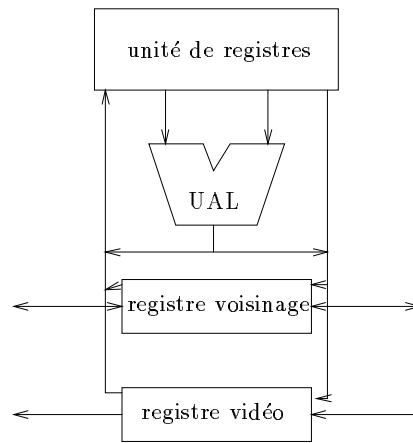


Figure 8 : Le processeur élémentaire de SLAP.

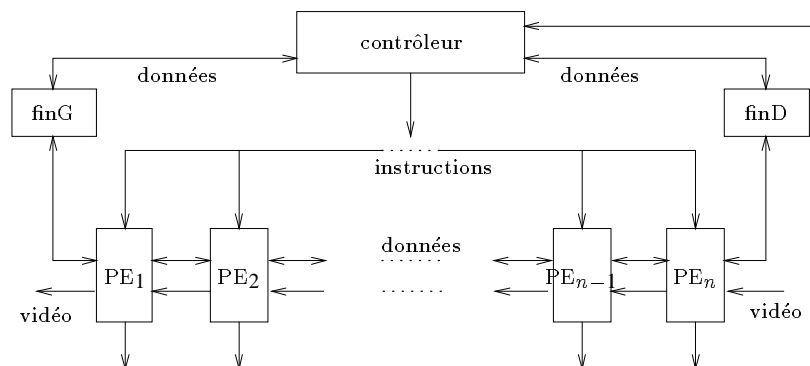


Figure 9 : Le système SLAP.

### 3.3 Machine AIS-5000 (Applied Intelligent Systems)

La machine commerciale AIS-5000 [ScWi88] intègre 1024 processeurs bit-série connectés linéairement et ayant chacun leur propre mémoire locale (32K bits).

Le processeur élémentaire permet trois types d'opérations : booléennes, arithmétiques, et des opérations de voisinage. Les opérations booléennes sont des opérations à 2, 3 ou 4 opérandes (lus en mémoire), la table de vérité est fournie par le programmeur. Les opérations de voisinage sont similaires aux opérations booléennes si ce n'est que les opérandes peuvent provenir de la mémoire des processeurs voisins. Les performances sont de l'ordre de 300 millions d'opérations par seconde sur des entiers 8 bits.

Un circuit précaractérisé intègre 8 processeurs élémentaires et leurs canaux d'entrée/sortie (3 canaux séparés pour différents échanges en parallèle). Une carte processeur consiste en 16 circuits (8 processeurs par circuit) et leurs mémoires associées (boîtier 32K octets). Un système à 1024 PE se compose de 8 cartes processeur.

Le système complet comporte le processeur hôte et ses périphériques (à base de MC68000), le réseau linéaire de processeurs fonctionnant en SIMD et son contrôleur, le module d'entrée/sortie spécialisées.

### 3.4 Machine SYMPATI 2 (CEA Saclay, IRIT Toulouse)

La machine SYMPATI 2 [JBEL88] consiste en un réseau linéaire de processeurs reliés en anneau (32 à 128 processeurs selon les configurations). Chaque processeur élémentaire a une mémoire locale de 32K octets.

Un circuit précaractérisé de 90 000 transistors (technologie CMOS 1,2  $\mu\text{m}$ , 224 broches) intègre 4 processeurs élémentaires à l'exception de la mémoire qui est externe. Une carte au format PC contient 32 processeurs élémentaires (8 boîtiers processeur) et 32 bancs mémoire.

Le processeur (figure 10) se compose de : une UAL 16 bits incluant les fonctions courantes (logiques, arithmétiques et décalages), un multiplieur 8 bits (résultat sur 16 bits), une unité de 6 registres accessibles par mot ou demi-mot, une unité d'adressage. L'unité d'adressage comporte un module de calcul d'adresse et un module de masquage permettant d'inhiber le traitement si le point traité est en dehors de la fenêtre considérée. L'unité de calcul d'adresse permet d'accéder aux bancs mémoire en utilisant, soit une adresse de base envoyée par l'unité de commande dans le cas du fonctionnement hélicoïdal, soit une adresse engendrée

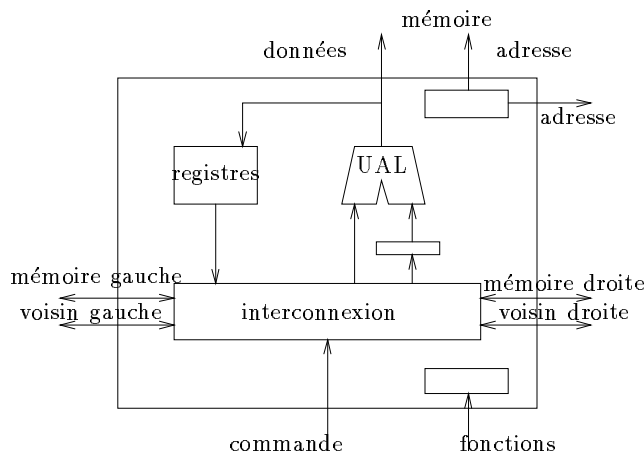


Figure 10 : Structure interne du processeur élémentaire de SYMPATI.

par la partie traitement du processeur dans le cas d'un fonctionnement tabulé (ces modes de fonctionnement sont décrits plus loin).

Des chemins de données permettent l'accès direct à un opérande :

- mémoire à une distance 0, 1 ou 2 à gauche ou à droite
- registre à une distance 0, 1, 2 ou 3 à gauche ou à droite comme illustré par la figure 11.

Toute instruction (accès à une donnée, calcul, stockage) s'effectue en un cycle.

Le nombre de processeurs peut être inférieur à la taille des images traitées. Les données sont distribuées selon un schéma hélicoïdal dans les bancs mémoire des processeurs de telle sorte qu'une lecture ou une écriture de  $n$  mots consécutifs d'une ligne ou d'une colonne de l'image puisse se faire sans conflit. La répartition hélicoïdale est telle que le processeur de rang  $k$  du réseau de  $n$  processeurs reçoit dans son banc mémoire les pixel  $(i, j)$  tels que  $(i + j) \bmod n = k$ , et ceci dans l'ordre rencontré en balayage ligne. L'image peut également être divisée en bandes, chaque bande de lignes ou de colonnes est affectée à un banc mémoire.

L'avantage de la répartition hélicoïdale des données est l'accès direct à un grand voisinage. Cette caractéristique est illustrée par la figure 12 où le pixel



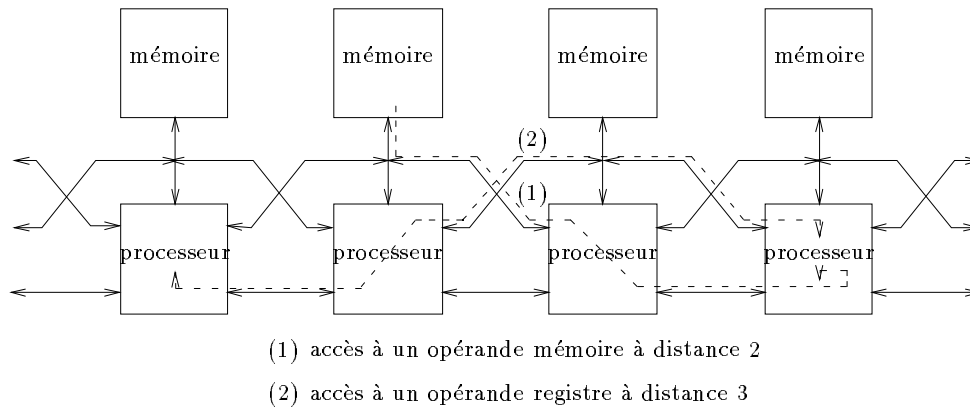


Figure 11 : Interconnexion des processeurs de SYMPATI.

central (représenté par un cercle) appartient au processeur  $k$ . Les autres pixels de cette partie d'image (symbolisés par un triangle) sont des pixels d'un voisinage  $7 \times 7$  et appartiennent aux bancs mémoire des processeurs voisins situés à distance au plus 6 (à gauche ou à droite) du processeur numéro  $k$ . En un cycle, tout pixel d'un voisinage  $3 \times 3$  est accessible puisque les interconnexions entre processeurs rendent possible un accès mémoire jusqu'à une distance 2.

Une unité de commande pilote le réseau de processeurs et assure l'interface avec l'hôte (PC) et les entrées/sorties vidéo. L'unité de commande est constituée d'un séquenceur, d'un module de balayage (propagation du traitement sur tout ou partie de l'image), d'un module d'entrées/sorties vidéo (acquisition/restitution de l'image), d'une interface PC (chargement micro-code, initialisation, etc). Les données sont stockées dans les bancs mémoire des processeurs à la cadence vidéo.

### 3.5 Machine PRINCETON (Centre de recherche SARNOFF)

La machine (figure 13) consiste en un réseau linéaire pouvant aller jusqu'à 2048 processeurs [Chin88] Chaque processeur (14Mhz, temps de cycle 70ns) contient un UAL 16 bits, un multiplieur 16 bits (résultat sur 32 bits), une unité de 64 registres 16 bits (3 ports : deux lectures, une écriture), une interface mémoire locale externe (16 bits adresses/données). La mémoire locale (externe au processeur) a

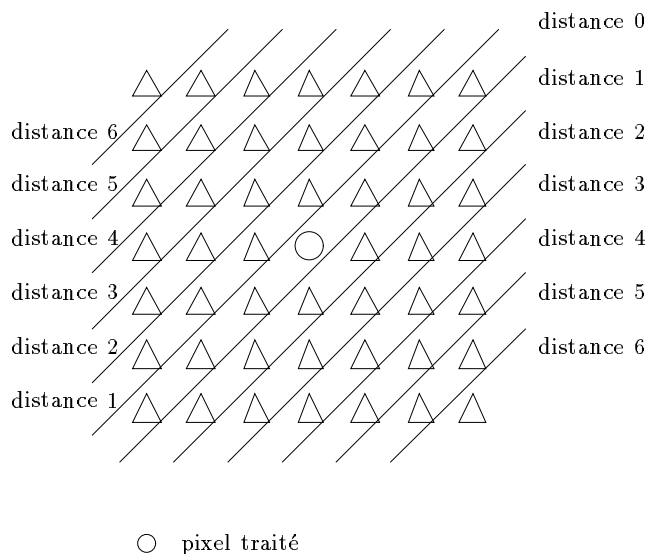


Figure 12 : Illustration de l'accès direct à un voisinage 3 x 3 dans SYMPATI.

une capacité de 16K mots de 16 bits, elle permet le stockage de 32 trames de données vidéo 16 bits (mémorisation d'une colonne d'image de taille 512 x 512 par processeur). La machine est contrôlée à chaque cycle d'horloge par une instruction de 89 bits (contrôle de la partie opérative et de l'unité de communication) fournie par l'unité de contrôle. La mémoire de microprogramme a une capacité de 16K instructions. L'instruction permet un parallélisme interne important (opération UAL, multiplication, accès mémoire).

Un bus de communication inter-processeur (IPC) (contrôlé par 16 bits) sert aux échanges entre processeurs, selon des principes voisins de ceux du Tore polymorphique : une opération de transfert de données est de type *diffusion* ou de type *bypass* :

- *mode diffusion* : un processeur émet vers plusieurs autres,
- *mode bypass* : établissement de liaisons bidirectionnelles entre processeurs distants sans recouvrement entre les liaisons.

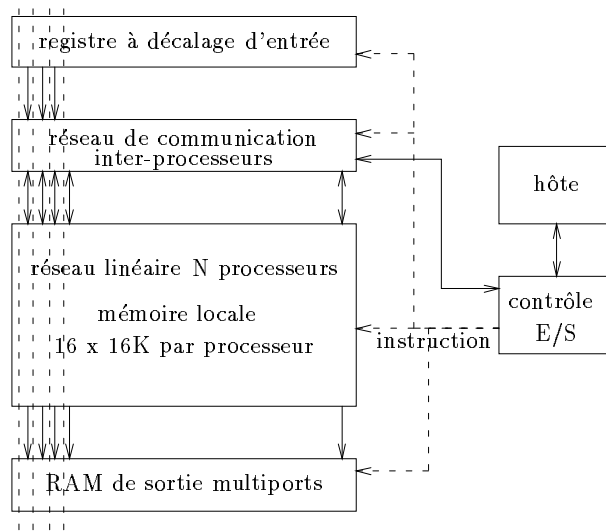


Figure 13 : Le système PRINCETON.

Quelque soit le mode de communication, le réseau permet les transferts entre processeurs dans une plage de 64 processeurs dans le cycle. Dans le pire des cas (réseau à 2048 processeurs), il faut 5 instructions pour effectuer le transfert. Ce réseau de communication est configuré dynamiquement à chaque cycle d'horloge, une nouvelle topologie peut être générée en 2 instructions.

Un système à 512 processeurs consiste en 8 cartes au format 22x17cm comportant chacune 32 circuits processeurs et 16 circuits d'entrées/sorties. Un circuit processeur intègre deux processeurs élémentaires (précaractérisé de 75000 portes, 223 broches).

## 4 Architectures pipelines

La structure pipeline peut s'avérer être intéressante pour un problème de traitement d'une série d'images sur chacune desquelles on doit réaliser la même séquence de traitement (enchaînements de tâches). Celle-ci consiste en une architecture multiprocesseur ou les processeurs sont linéairement connectés, un flot d'images

transite à travers le réseau (la sortie du premier processeur est connectée à l'entrée du second, etc). En mode pipeline, le temps critique est celui de l'étage le plus lent du pipeline.

Différentes réalisations illustrent ce mode de fonctionnement :

- La machine Cytocomputer (université du Michigan)
- La machine PIPE

#### 4.1 Machine Cytocomputer (université du Michigan)

La machine Cytocomputer [LoMc80], illustrée par la figure 14, consiste en une pipeline unidirectionnel d'étages de traitement ayant tous une horloge commune. Chaque étage du pipeline réalise une transformation élémentaire sur l'image (fonction logique de voisinage). Les images entrent dans le pipeline comme un flot de pixels au format balayage ligne et progressent dans le pipeline à un rythme constant. A l'issue de la phase d'initialisation du pipeline, les images calculées sortent au rythme des images d'entrée. Chaque étage comprend un ensemble de registres à décalage capable de mémoriser deux lignes contiguës d'une image tandis qu'une fenêtre de registres mémorise les 9 pixels d'un voisinage qui constituent la fenêtre d'entrée 3 x 3 du module logique de voisinage. Ce module réalise la transformation programmée du pixel central du voisinage à partir des valeurs du pixel central et de ses huit voisins (figure 15).

#### 4.2 Machine PIPE

La machine est un réseau linéaire de processeurs identiques [KeSL85] [VeMD86], dont les extrémités sont connectées à des processeurs d'entrée-sortie eux mêmes connectés à une station hôte. Chaque processeur possède son contrôleur chargé par l'hôte. Un PE est constitué de 2 cartes.

Deux types de traitements sont réalisés par les PE : des opérations arithmétiques ou booléennes sur des pixels, et des opérations de voisinage (3 x 3). Les chemins de données externes sont de 8 bits, mais en interne la taille des opérandes est augmentée de manière à éviter les pertes de précision. Les nombreuses fonctions du processeur doivent permettre le traitement en une unité de temps (1/60s), temps pendant lequel les images 256 x 256 transitent sur les ports de communication.

Chaque processeur est censé effectuer un ensemble d'opérations pendant cette unité de temps sur l'ensemble des pixels d'une(des) image(s). Le PE a 3 entrées

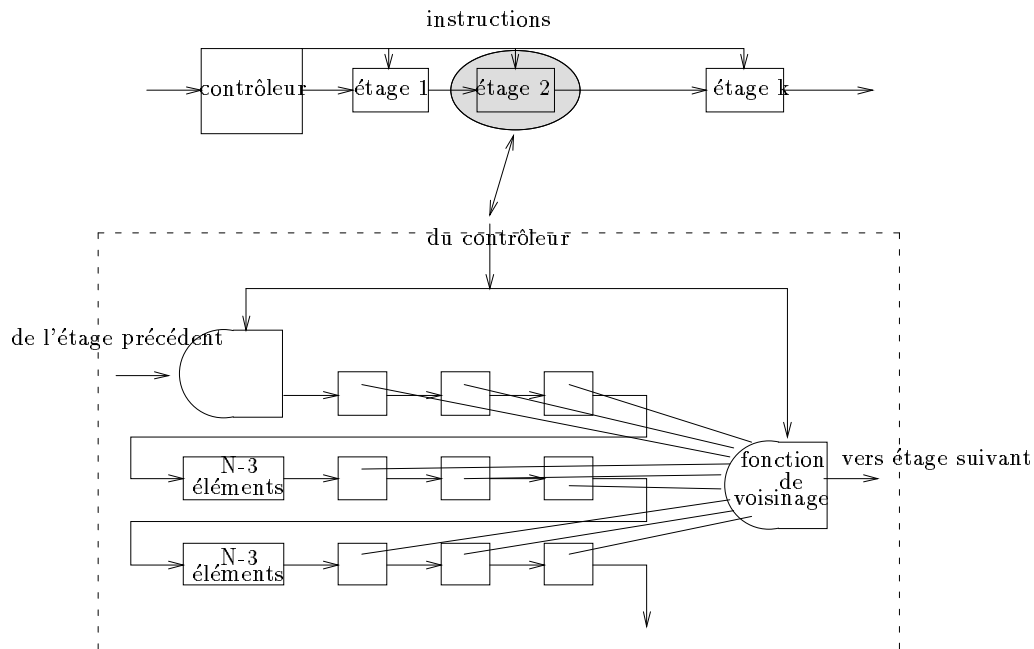


Figure 14 : Structure pipeline Cytocomputer.

appelée *forward*, *backward* et *recursive* et 3 sorties (voir figure 16) et est connecté à 2 bus généraux. Le chemin de données *forward* permet d'implémenter une structure pipeline classique. Le chemin *recursive* permet de simuler un pipeline de longueur quelconque avec un étage et permet la mise en œuvre d'algorithmes nécessitant plusieurs itérations avant de converger (relaxation ou simulation de voisinages importants par application de plusieurs opérations sur un petit voisinage). Le chemin *backward* permet d'effectuer aussi bien des opérations de voisinage spatial que temporel.

La figure 17 présente la structure interne d'un processeur de la machine PIPE. Chacune des entrées peut être traitée séparément (opération de seuillage, opération arithmétique ou booléenne), celles-ci peuvent ensuite être combinées entre elles (UAL), avec des poids différents, pour donner une image unique. Cette image ainsi que celles en provenance d'un canal DMA peuvent être stockées dans l'un ou (et)

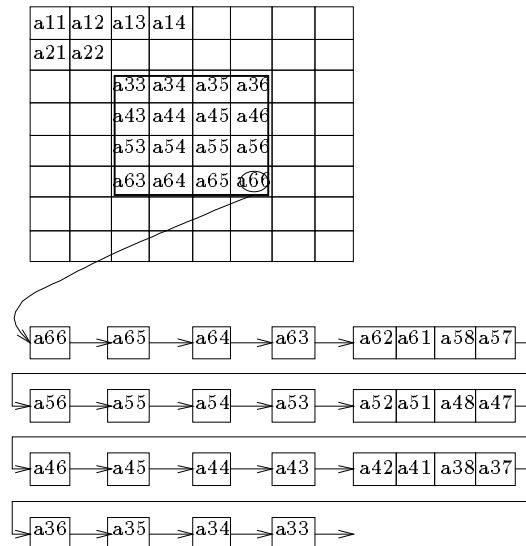


Figure 15 : Décalage des données dans la machine Cytocomputer.

l'autre des 2 tampons d'image (sorties X et Y). X ou Y passent ensuite dans une fonction arbitraire (look-up table) et un générateur de voisinage (GV). X, Y peuvent ensuite être envoyés dans 2 opérateurs de voisinage NOP (convolution arithmétique ou jeu d'opérateurs booléens). Les sorties de NOP attaquent un générateur de fonction (look-up table) (sortie U) et une UAL (sortie V).

Les 3 sorties d'un PE peuvent être choisies parmi X, Y, U ou V chacune. L'opérateur de voisinage fonctionne en mode pipeline.

A l'intérieur d'un PE un mécanisme de retard (mis en œuvre par action sur l'adressage) permet d'assurer la cohérence des traitements de voisinage lors de l'utilisation de l'entrée *recursive*. Un module de résolution permet lors des accès aux tampons de travailler sur une image 4 fois plus petite ou 4 fois plus grandes (par action sur adresses en écriture ou lecture des tampons d'image).

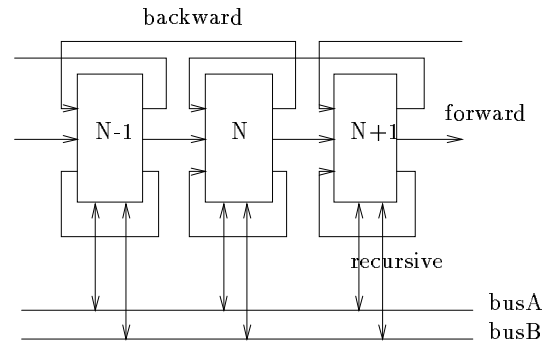


Figure 16 : communication entre étages dans PIPE.

## 5 Architectures pyramidales

Le volume considérable des données mises en jeu lors de la manipulation d'une image justifie l'utilisation d'un grand nombre de processeurs. Les réseaux linéaires et matriciels sont bien adaptés au traitement d'images de bas niveau. Pour des traitements de plus haut niveau, certains considèrent le manque de communications globales entre processeurs comme un inconvénient majeur (plus particulièrement pour les opérations mettant en jeu un grand nombre de pixels en analyse d'images). Une des approches proposées pour pallier cet inconvénient repose sur l'organisation pyramidale des communications qui permet d'effectuer rapidement les opérations de concentration et de diffusion d'informations. Plusieurs pyramides sont en cours d'étude (SPHINX (Orsay), PAPIA (Italie), GAM (USA [Uhr87]).

La plupart des architectures pyramidales proposées reposent sur une pyramide quaternaire c'est à dire sur une interconnexion où le sommet de la pyramide est la racine d'un arbre quaternaire. Une pyramide binaire présente un coût supérieur par rapport à une pyramide quaternaire dont la taille de la base (dernier étage) serait identique : le nombre de processeurs est de 50% plus élevé, le nombre d'étages est doublé, le nombre de connexions entre processeurs est supérieur de 15%.

Un exemple de machine pyramidale, la machine SPHINX, développé à Orsay est détaillée.

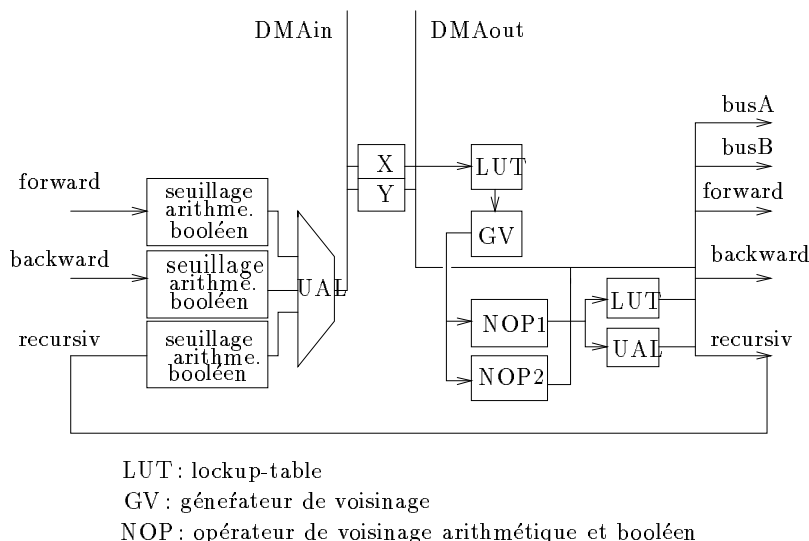


Figure 17 : architecture d'un PE de PIPE.

## Machine SPHINX(Orsay)

SPHINX [MeND91] est formé d'un ensemble de processeurs élémentaires interconnectés selon une architecture en pyramide. La pyramide a une hauteur  $h$  (nombre d'étages). Tous les processeurs d'un même étage sont interconnectés selon une maille carrée à deux dimensions. Les processeurs de deux étages consécutifs de la pyramide sont interconnectés de manière à ce que le sommet de la pyramide soit la racine d'un arbre binaire.

Le contrôle de la machine est multi-SIMD : chacun des étages fonctionne en mode SIMD mais chaque étage dispose de son propre contrôleur (mode MIMD entre étages). Ce mode de fonctionnement impose de pouvoir synchroniser les étages lors d'échanges d'information entre eux en un temps de l'ordre de grandeur du temps d'exécution d'une instruction sur les processeurs élémentaires. Le problème est résolu en distinguant deux aspects de nature très différente dans le contrôle. D'une part, chacun des étages fonctionne sous contrôle de son propre programme chargé d'assurer un séquençement correct des instructions. Les pro-



grammes exécutés par les différents étages ne sont pas indépendants, ils coopèrent de manière à assurer une manipulation cohérente de l'information par la structure. Cette coopération se traduit par l'échange de paramètres entre étages. Il s'agit d'un mode de synchronisation à grain suffisamment gros pour qu'il puisse se faire par des mécanismes classiques de communication sans surcoût notable (contrôle de haut niveau). D'autre part, il faut assurer l'intégrité de l'information échangée entre étages (contraintes de type synchronisation producteur consommateur). Ces contraintes impliquent uniquement des étages voisins et sont résolues par un mécanisme câblé contrôlant l'état du tampon de communication inter-étages au regard des propriétés de communication des instructions exécutées par chacun des étages (contrôle de bas niveau). Chacun des modes est assuré par un organe matériel spécifique. Le contrôleur de haut niveau est réalisé par un microprocesseur traditionnel : sa fonction est de gérer les instructions émises à la pyramide en exécutant un programme utilisateur compilé, la mémoire des processeurs et les entrées sorties. Les différents contrôleurs sont interconnectés par une liaison point à point et par une mémoire partagée. Les instructions produites par un macro-générateur matériel dédié (transformation des instructions portant sur des mots en des instructions bit-série) sont transmises vers les contrôleurs de bas niveau qui les synchronisera avant émission vers la pyramide.

Le processeur élémentaire comprend une mémoire interne de 256 bits, une unité arithmétique et logique et un certain nombre de registres dédiés. L'ensemble des traitements et des communications est réalisé en bit-série. Un circuit intégrant 16 PE a été réalisé (technologie 1,5  $\mu\text{m}$ , 100K transistors, boîtier PGA 100 broches, fréquence de fonctionnement de 14,3MHz). Une addition de deux mots de 8 bits est réalisée en 2,5  $\mu\text{s}$ , une multiplication en 18  $\mu\text{s}$ .

## 6 Machines MIMD pour le traitement d'image

Dans une machine MIMD chaque processeur est entièrement programmable et exécute son propre programme. Ces machines sont plus flexibles que les précédentes, elles permettent la mise en œuvre de différents parallélismes.

Quelques exemples de machines utilisées en traitement d'image sont brièvement décrits.

subsectionMachine PASM (université de Purdue)

PASM [Sieg81] est un système multiprocesseur conçu et développé à l'université de Purdue par Siegel pour le traitement d'image. L'aspect intéressant du système est sa reconfigurabilité : les études dans ce domaine sont peu nombreuses.

Le système peut être configuré dynamiquement pour opérer comme un ensemble de machines SIMD (multi-SIMD) et/ou comme une machine MIMD.

La machine se compose des modules suivants :

- L'unité de calcul parallèle est constituée de  $N = 2^n$  processeurs, de  $N$  modules mémoires et d'un réseau d'interconnexion. Les processeurs sont des microprocesseurs qui fonctionnent en SIMD ou MIMD. Le réseau d'interconnexion permet les échanges de données entre les processeurs et les mémoires ;
- Un ensemble de microcontrôleurs agissent en tant qu'unités de contrôle des processeurs de l'unité de calcul parallèle. Il y a  $Q = 2^q$  microcontrôleurs. Chaque microcontrôleur contrôle  $N/Q$  processeurs de l'unité de calcul. A chaque microcontrôleur est associé un module mémoire. Ces microcontrôleurs permettent à PASM d'opérer comme un système multi-SIMD.

## 6.1 Machine WARP (université de Carnegie-Mellon)

La Machine WARP [KuWe86], [Anna87] est une machine systolique programmable développée par l'université de Carnegie Mellon entre 1984 et 1986 et commercialisée par General Electric en 1987. L'organisation de la machine WARP se compose de trois parties : l'hôte, l'unité d'interface et un réseau linéaire de 10 processeurs relié à l'unité d'interface par les processeurs extrêmes. L'hôte communique avec l'interface via un bus 32 bits. L'interface dispose de piles d'entrées/sorties pour les échanges avec l'hôte, d'un mécanisme de conversion entier/flottant pour accroître la bande passante entre l'hôte (possibilité de compaction de données lors des communications) et le réseau et de facilités pour la génération des adresses d'accès aux mémoires des processeurs. Une machine WARP a une performance de 100 MFLOPS (millions d'opérations flottantes par seconde).

Chaque processeur est contrôlé par un microprogramme à structure horizontale et possède son propre microséquenceur et sa propre mémoire de programme. Les chemins de données du processeur WARP sont composés d'un multiplieur, d'un additionneur (tous deux d'une puissance de 5 MFLOPS), d'une mémoire locale et de deux files d'entrées/sorties pour les canaux de communication. Tous ces éléments

sont reliés entre eux par une matrice de points de croisement (*crossbar*) (figure 18). La bande passante en entrées/sorties est très importante, 80 Méga-octets par seconde, pour permettre la circulation des données entre les processeurs.

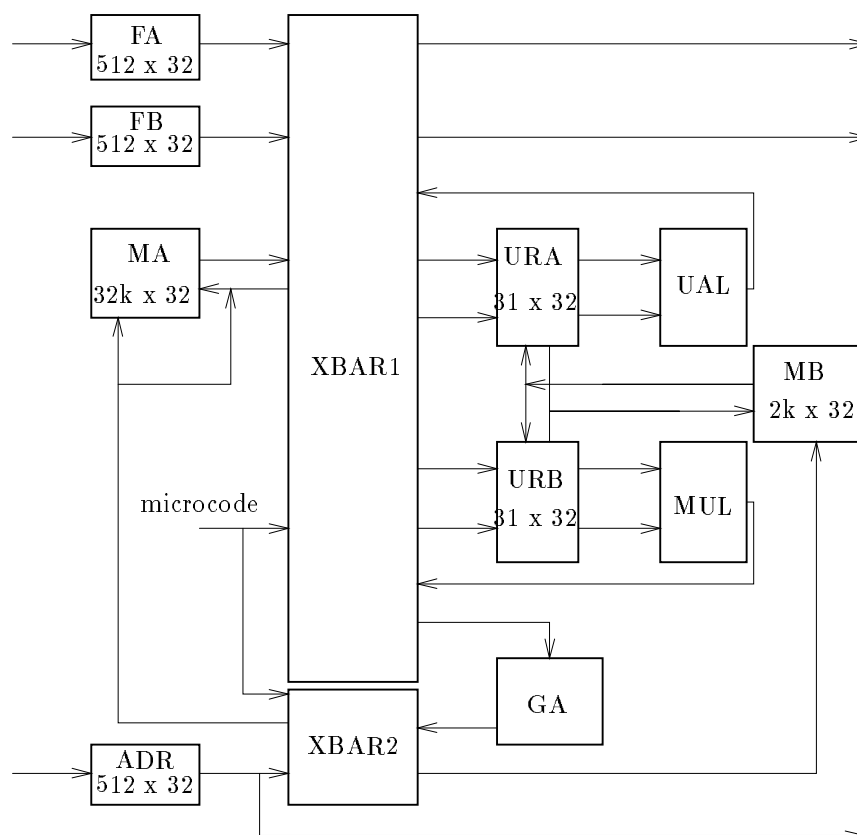
## 6.2 Machine iWARP (université de Carnegie-Mellon et Intel)

Le iWARP, développé conjointement par Intel et l'université de Carnegie-Mellon [Bork88] et financé par la DARPA durant quatre années, est une version intégrée du processeur élémentaire de la machine WARP. Le iWARP utilise l'approche VLIW (*Very Long Instruction Word*), c'est à dire que des instructions longues spécifient explicitement l'ensemble des opérations s'exécutant dans une même instruction. Un tableau de 64 processeurs iWARP pourra atteindre une puissance de l'ordre du GFLOPS.

Le processeur iWARP intègre toute les fonctionnalités du processeur WARP sur un seul circuit intégré à l'exception de la mémoire. Le circuit comporte 650 000 transistors, il atteint une puissance de 20 MFLOPS et est cadencé par une horloge à 40 MHz [Pete90].

Le composant iWARP, dont l'organisation est donnée figure 19, consiste en trois unités autonomes et fortement connectées.

- L'unité de calcul est organisée autour d'une unité de 128 registres 32 bits multiports (15 ports), elle atteint une performance de 20 MFLOPS en simple précision (arithmétique virgule flottante 32 bits) et 10 MFLOPS en double précision (64 bits). L'unité de calcul est constituée d'une UAL et d'un multiplieur en arithmétique flottante ainsi que d'une unité arithmétique et logique opérant sur 8, 16 ou 32 bits. Les opérations de division et de racine carrée sont également supportées. L'unité de registres supporte jusqu'à neuf opérations de lecture et six opérations d'écriture dans un cycle d'horloge de 50 ns.
- L'unité de communication [Bork90] implémente les mécanismes d'échange entre processeurs iWARP. Les 8 bus d'entrées/sorties (4 en entrée, 4 en sortie) sur une largeur de 8 bits chacun, supportent une bande passante égale à 320 méga-octets par seconde. Chaque port physique peut être multiplexé pour offrir jusqu'à 20 chemins de communication. Un service de routage de type *wormhole*, comme pour les hypercubes de deuxième génération, est mis en œuvre. Le message contient une information sur le destinataire, celle-ci



UAL, MUL : unités arithmétiques  
 FA, FB, ADR : files d'entrée  
 URA, URB : unités de registres  
 MA, MB : bancs mémoire  
 GA : générateur d'adresse  
 XBAR1 : réseau d'interconnexion de données  
 XBAR2 : réseau d'interconnexion d'adresses

Figure 18 : Chemin de données du processeur WARP.

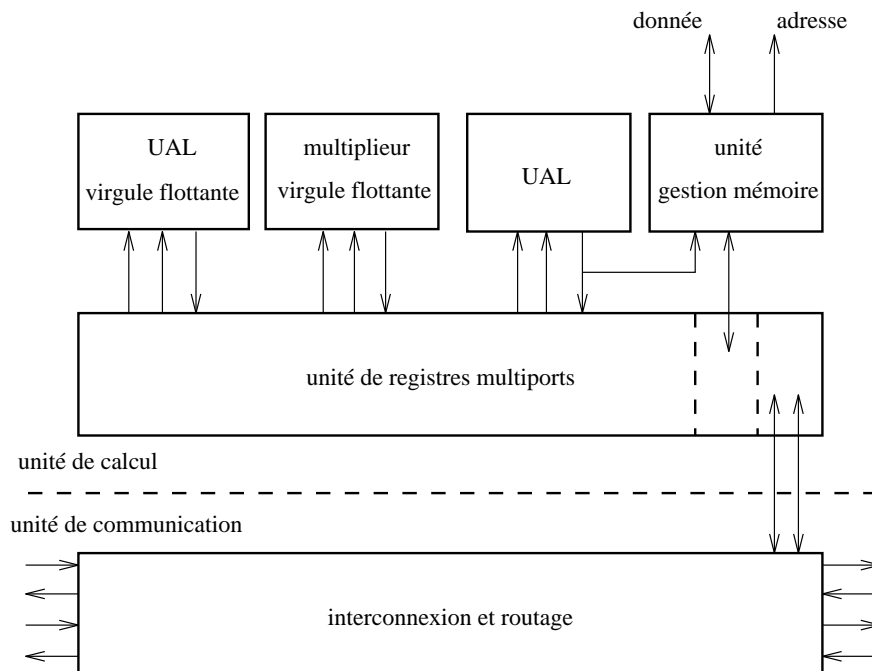


Figure 19 : Organisation du circuit iWARP.

est lu “au vol” par le module de communication et permet la sélection en temps réel du prochain port de sortie (si besoin). Le contenu du message est transmis mot par mot au destinataire ; si un port de sortie n’est pas disponible, la progression du message s’arrête et redémarre lorsque le port devient libre.

- L’unité d’échange mémoire fournit une interface à une mémoire locale externe avec une bande passante égale à 160 méga-octets par seconde (bus de données 64 bits, bus d’adresse 24 bits).

Intel a développé deux prototypes d’architecture à base de processeur iWARP, leurs commercialisations ont été annoncées au mois d’août 1991. Le premier prototype consiste en un ensemble de cartes (*Single Board Array, (SBA)*) pouvant être intégrées dans une station de travail SUN. Chaque SBA est constitué de quatre processeurs iWARP organisés en tableau de  $2 \times 2$  processeurs dotés chacun d’une mémoire locale dont la taille peut varier entre 0.5 et 4 Mo selon la configuration choisie. Une station SUN peut supporter jusqu’à 8 cartes SBA permettant ainsi la

réalisation d'un tableau de processeurs de  $2 \times 16$  iWARP pour une performance maximale de 640 MFLOPS. Le deuxième prototype se présente sous la forme d'un ensemble d'armoires (4 maximum, 1024 processeurs) similaires à celles utilisées pour les architectures iPSC. Dans les deux cas, la topologie de l'architecture est une grille 2D. Le réseau de processeurs iWARP est connecté à une station de travail SUN grâce à une ou plusieurs cartes d'interface.

### 6.3 Machine GPIIP (General Purpose Image Processor, HITACHI)

HITACHI a développé une machine de traitement d'image très générale organisée autour de 64 circuits DSP [Head88]. Un DSP spécifique a été développé (temps de cycle 50ns, 430K transistors,  $150\text{mm}^2$ , technologie CMOS  $1.3 \mu\text{m}$ ).

Le DSP (16 bits) développé se différencie des DSP actuels par sa hiérarchie d'instructions à deux niveaux : les microinstructions pour les traitements internes, et les picoinstructions pour les entrées/sorties externes, son organisation mémoire interne (4 pages de 512 mots de 16 bits) et son interface multi-processeur. Une microinstruction contrôle jusqu'à 8 ressources en parallèle : UAL ou opérateur booléen, multiplieur, 3 bus interne et 3 unités de calcul d'adresse.

La machine GPIIP se compose de 64 processeurs. Chaque processeur consiste en 1 DSP et sa mémoire locale externe de 512K octets. Une unité de contrôle de chemin, située entre le DSP et sa mémoire locale permet la liaison entre processeurs voisins, la liaison ainsi formée constitue un pipeline en anneau. Cette liaison contrôlée par une unité de gestion de l'anneau fournit un mécanisme pour l'échange de données image entre processeurs, la diffusion d'une même image à l'ensemble des processeurs, la circulation de résultats intermédiaires entre les processeurs.

Le système complet consiste en 8 cartes processeurs, une carte contrôleur et une carte entrée/sortie image. Chaque carte processeur ( $45 \times 43\text{cm}$ ) comporte 8 DSP.

### 6.4 Machine NEC

NEC a développé un système multiprocesseur pour le traitement temps réel d'images de télévision haute définition (TVHD) [Tami89] [TaHN89]. L'approche proposée consiste en un ensemble de 128 processeurs de traitement de signal vidéo (VSPM) répartis en 8 groupes de 16 processeurs chacun. Les groupes sont reliés aux convertisseurs analogique/numérique ou numérique/analogique par l'intermédiaire d'une

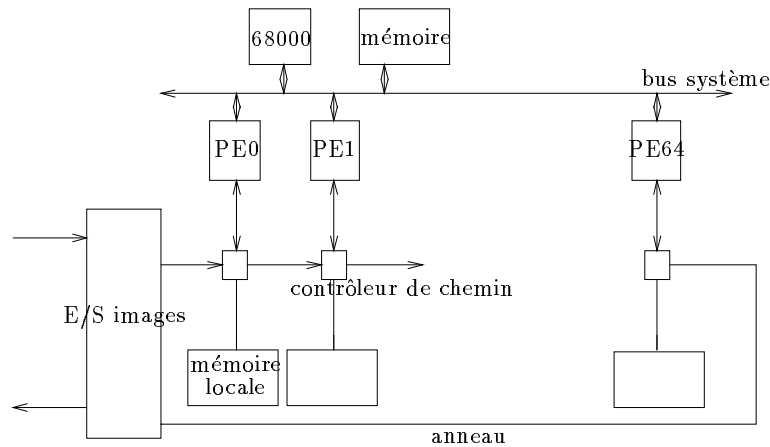


Figure 20 : Le système GPIIP.

matrice de points de croisement. Des groupes voisins peuvent être connectés en parallèle et/ou en pipeline sous le contrôle d'une unité de commutation de bus (BSU) associée à chaque groupe (figure 21).

### Architecture du VSPM

Le VSPM est un processeur programmable 16 bits et consiste en 4 modules : les unités de stockage d'entrée et de sortie, l'unité de traitement et l'unité de contrôle. L'unité de stockage d'entrée se compose de deux mémoires (64K mots de 16 bits) et leurs unités de génération d'adresse associées. L'unité de sortie consiste en deux FIFOs (64K mots de 16 bits). L'unité de traitement se compose d'une unité arithmétique pipelinée, de deux mémoires internes (64K mots de 16 bits) et leurs unités de génération d'adresse associées, de l'unité de séquencement et de la mémoire de programme (16K mots de 89 bits), 3 bus internes permettent les échanges entre ces différentes unités. Un VSPM (20 MOPS) consiste en une carte au format 30x30cm.

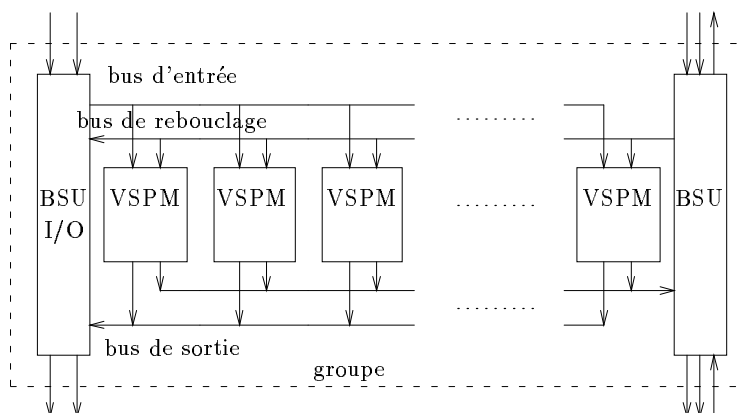


Figure 21 : Machine NEC.

## 7 Discussion

Comme l'illustre la présentation faite dans cet article, pour la réalisation d'architectures parallèles spécialisées adaptée au traitement de l'image, plusieurs approches sont proposées. Dans la catégorie des machines SIMD, on note les réseaux bidimensionnels à base de processeurs bit-série (MPP, GAPP, DAP, etc), les réseaux linéaires réalisés à partir de processeurs plus performants (16 voire 32 bits), (CLIP7, SLAP, SYMPATI, PRINCETON). Parmi les machines MIMD, on note essentiellement les machines WARP, iWARP, et les approches à base de processeurs de traitement de signal (DSP) supportant ainsi différents types de parallélisme (systolique, pipeline, ...). Nombre de structures décrites ici ont été conçues dans les laboratoires de recherche, elles n'ont, qu'à quelques rares exceptions près, jamais quitté le stade de prototype en dépit parfois de concepts architecturaux très évolués. L'absence d'un réel marché en est la raison principale.

Un nombre restreint de systèmes se partagent, à l'heure actuelle, le marché. Ceux-ci sont généralement constitués d'une ou de plusieurs cartes processeur, de mémoires, d'un dispositif d'acquisition d'images. Ils comportent généralement des cartes accélératrices, dédiées au traitement de bas et de moyen niveau, réalisées à partir d'opérateurs spécifiques. Tous ces éléments sont reliés à un bus système,



auquel on adjoint si nécessaire des chemins de communication spécifiques (bus vidéo, liaison pipeline). L'ensemble est contrôlé par la carte mère du système.

Cependant, l'émergence de nouveaux domaines d'applications comme par exemple la compression numérique d'images (systèmes de visiophonie et de vidéoconférence, stockage de bases de données images, magnétoscopes numériques haute qualité, télévision haute définition) qui représentent des marchés stratégiques pour les télécommunications, la bureautique ou les applications grand public, contribueront, très certainement, à une ouverture du marché.

### **Approches SIMD ou pipelines?**

Une comparaison des différentes machines est difficile car les critères sont nombreux : taille, coût, performance, contexte applicatif, facilités d'utilisation, etc. Néanmoins, il est intéressant de constater que deux styles d'architectures prédominent : les architectures SIMD tels les tableaux de processeurs et les architectures de type pipeline.

Les architectures SIMD sont bien adaptées aux traitements de bas et de moyen niveau, caractérisés par l'application d'opérations identiques à tous les pixels. Les approches bit-série sont à l'heure actuelle encore privilégiées. Un niveau de parallélisme très fin peut être obtenu avec ce type de processeur. Par contre, la programmation d'une telle machine est assez compliquée lorsque des traitements sur des données de 16 ou 32 bits sont nécessaires étant donné que les traitements se font en bit-série.

Les architectures pipelines sont plus spécialisées et offrent moins de possibilités que les architectures SIMD. Elles correspondent plus à la notion de coprocesseurs adaptés à une classe de traitements (filtrage, transformée FFT, DCT, etc) et délivrent des performances élevées pour une mise en œuvre généralement très simple. Les données sont fournies une à une au coprocesseur et les résultats sont récupérés au même rythme. Ces opérateurs peuvent être généralement placés en cascade afin d'accroître les performances.

Les architectures SIMD n'ont pas rencontré le succès commercial attendu même si leur polyvalence et les performances élevées qu'elles délivrent en font de véritables supercalculateurs : citons pour exemple la Connection Machine [Hill85, TuRo88], et les machines MasPar toutes récentes sur le marché [Blan90, Nick90]. Ces structures, bien que généralement simples et régulières conduisent à des réalisations très lourdes et très coûteuses n'exploitant finalement que partiellement les possibilités offertes par les techniques d'intégration actuelle. Le parallélisme

mis en œuvre repose sur un découpage des données en fragments répartis dans des mémoires associées à chaque processeur élémentaire. Deux solutions alors peuvent être considérées.

- Mémoires et processeurs sont placés sur le même substrat et dans ce cas, seul un petit nombre de processeurs sont intégrés sur un circuit. Les technologies habituellement utilisées pour la réalisation de mémoire ne peuvent être employées.
- Processeurs et mémoires consistent en des circuits différents. Des mémoires du commerce sont alors utilisées. Bien que très denses, ces mémoires dont les chemins d'accès sont généralement limités (1 à 4 par boîtier), ne se prêtent pas bien à un nombre élevé d'accès simultané. Le recours à un nombre élevé de mémoires de petite taille est une barrière à une intégration poussée.

Par ailleurs, l'idée de base qui consiste à associer un processeur élémentaire par pixel de l'image peut s'avérer être trop coûteuse en matériel. Aussi, le nombre de processeurs est toujours inférieur à la dimension de l'image, ce qui oblige à découper les données en blocs et rend la mise en œuvre de l'application plus complexe.

## Évolution des architectures pour l'image

Les applications de traitement d'image impliquent des algorithmes de plus en plus complexes. Il ne s'agit plus uniquement de mettre en œuvre un traitement ou un algorithme de base mais de maîtriser toute une application faite d'un certain nombre de traitements de base qui interagissent. La mise au point, l'émulation et l'évaluation de ces applications complexes nécessitent des architectures parallèles spécialisées qui doivent être :

- *programmables et polyvalentes* pour pouvoir effectuer les algorithmes souhaités et s'adapter aux évolutions prévisibles dans le domaine des algorithmes ;
- *modulaires*. La modularité est une garantie sur les possibilités d'évolution de la machine et permet une adaptation aisée à des contextes applicatifs différents (adaptation à la complexité du type de problème à traiter).

La définition d'une architecture parallèle adaptée au traitement d'image implique l'examen de différents aspects en lien étroit avec le domaine d'applications visé : style de fonctionnement, topologie du réseau de processeurs, structure du processeur, dimensionnement de l'architecture, programmation.

Les styles de fonctionnement SIMD, pipeline, multi-SIMD et même MIMD peuvent être considérés. Force est de constater que les machines capables de supporter efficacement ces différents modes sont, à l'heure actuelle, peu nombreuses. La machine iWARP est à l'heure actuelle celle qui s'y prête le mieux. La machine NEC décrite dans le paragraphe 6.4 tente également d'offrir un aspect configurabilité. Notons également que des études d'évaluation de structures configurables en SIMD et/ou MIMD sont menées au sein de l'équipe SYMPATI à l'Irit à Toulouse [HoBL91].

Dans le cas du fonctionnement totalement SIMD, une organisation linéaire a des avantages sur des organisations de type grille 2D, pyramide ou hypercube : un nombre quelconque de processeurs peuvent être intégrés sur un même circuit sans modification de son brochage. De plus, les interconnexions entre circuits et cartes sont plus simples pour un réseau linéaire. Dans le cas de structures configurables (pipeline, multi-SIMD, systolique), les organisations peuvent être très diverses comme l'atteste les nombreux travaux de recherche sur ce thème. Il est à noter que l'évolution de la technologie permet de nos jours, l'intégration de quelques millions de transistors sur un circuit. Dans quelques années, c'est de dizaines voire d'une centaine de millions de transistors dont il sera question. Cette évolution aura très certainement un impact sur la topologie, l'organisation et le fonctionnement des machines spécialisées.

La plupart des processeurs spécialisés pour l'image conçus à ce jour sont des processeurs bit-série. Or les images sont rarement binaires. Cette approche est généralement justifiée par le fait que ces processeurs s'intègrent souvent dans des structures multidimensionnelles et que le nombre de processeurs peut être ainsi très important. Cependant, cette approche bit-série, compliquant la programmation, n'a plus de raison d'être si ce n'est dans des circuits dédiés, les technologies actuelles et à venir autorisant en effet une intégration massive. La tendance actuelle est plutôt de concevoir ou d'utiliser des processeurs puissants. Un certain nombre de "briques de base" devant permettre la réalisation de structures parallèles spécialisées sont maintenant étudiées et proposées par les industriels [Char93].

La définition d'une architecture parallèle ne peut plus être réalisée sans que sa programmation soit prise en compte, et ce, dès sa phase de définition. A l'heure actuelle, seules les machines SIMD, disposent d'environnements de programmation évolués permettant de tirer réellement parti des performances de la machine.

Il n'en est pas de même des architectures MIMD, pour lesquelles les outils de programmation (parallélisation, problèmes de placement/ordonnancement) ne sont pas aussi avancés.

## References

- [Adam89] P. Adam, D. Juvin, H. Essafi, P. Fernandez, et J.L. Basille. 4LP - Low Level Language for the Line Processor SYMPATI 2. In *12 Colloque GRETSI - Juan-Les-Pins*, pages 873–876, juin 1989.
- [Anna87] M. Annaratone, E. Arnould, T. Gross, H.T. Kung, M. Lam, O. Menzicoglu, et J. Webb. The Warp Computer: Architecture, Implementation and Performance. *IEEE Transactions on Computers*, C-36(12):1523–1538, décembre 1987.
- [Batc80] K.E. Batcher. Design of a Massively Parallel Processor. *IEEE Transactions on Computers*, C-29(9):836–840–349, septembre 1980.
- [BDHR90] D.W. Blevins, E.W. Davis, R.A. Heaton, et J.H. Reif. BLITZEN: A Highly Integrated Massively Parallel Machine. *Journal of Parallel and Distributed Computing*, 8(2):150–160, 1990.
- [Blan90] T. Blank. The MasPar MP-1 Architecture. In *Proceedings of IEEE Compton Spring 1990*, pages 20–24, IEEE, février 1990.
- [Bork88] S. Borkar, R. Cohn, G. Cox, T. Gross, H.T. Kung, M. Lam, M. Moore, C. Peterson, J. Pieper, J. Rankin, P.S. Tseng, J. Sutton, J. Urbanski, et J. Webb. iWarp: An Integrated Solution to High-Speed Parallel Computing. In *Proceedings of Supercomputing '88*, pages 330–339, Orlando FL (USA), novembre 1988.
- [Bork90] S. Borkar, R. Cohn, G. Cox, T. Gross, H.T. Kung, M. Lam, M. Levine, B. Moore, W. Moore, C. Peterson, J. Susman, J. Sutton, J. Urbanski, et J. Webb. Supporting Systolic and Memory Communication in iWarp. In *Proc. 17th Annual Symposium on Computer Architecture*, pages 70–81, Seattle WA (USA), mai 1990.
- [Char93] F. Charot. *Briques de base pour la réalisation d'architectures parallèles spécialisées*. Publication interne 721, IRISA, avril 1993.

- [Chin88] D. Chin, J. Passe, F. Bernard, H. Taylor, et S. Knight. The Princeton Engine: A Real-Time Video System Simulator. *IEEE Transactions on Consumer Electronics*, 34(2):285–297, mai 1988.
- [Duff78] M.J. Duff. Review of the CLIP Image Processing System. In *Proc. National Computer Conference*, pages 1055–1060, juin 1978.
- [FiHR87] A.L. Fisher, P.T. Highman, et T.E. Rockoff. Architecture of a VLSI SIMD Processing Element. In *Proc. IEEE conference on Computer Design: VLSI in Computers and Processors.*, pages 324–327, octobre 1987.
- [FiHR88] A.L. Fisher, P.T. Highnam, et T.E. Rockoff. Scan Line Array Processors: Work in Progress. In *Proc. Image Understanding Workshop*, pages 625–633, 1988.
- [Fish86] A.L. Fisher. Scan Line Array Processors for Image Computation. In *Proc. 13th Annual International Symposium on Computer Architecture*, pages 338–345, juin 1986.
- [FoMD88] T.J. Fountain, K.N. Matthews, et M.J. Duff. The CLIP7A Image Processor. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(3):310–319, mai 1988.
- [Head88] H. Heada, K. Kato, H. Matsushima, K. Kaneko, et M. Ejiri. A Multiprocessor System Utilizing Enhanced DSP's For Image Processing. In *Proc. International Conference on Systolic Arrays*, pages 611–620, mai 1988.
- [Hill85] W.D. Hillis. *The Connection Machine*. MIT Press, Cambridge, MA, 1985.
- [HoBL91] D. Houzet, J.L. Basille, et J.Y. Latil. évaluation du multiprocesseur SIMD/MIMD GFLOPS sur des algorithmes de traitement d'image. In *treizième Colloque GRETSI - Juan-Les-Pins*, pages 1197–1200, juin 1991.
- [JBEL88] D. Juvin, J.L. Basille, H. Essafi, et J.Y. Latil. Sympati2, a 1.5d processor array for image applications. In *EUSIPCO Signal Processing IV: theories and application*, pages 311–314, North Holland, 1988.

- [KeSL85] E.W. Kent, M.O. Shneier, et R. Lumia. PIPE (Pipelined Image Processing Engine). *Journal of Parallel and Distributed Computing*, 2:50–78, 1985.
- [Kung88] S.Y. Kung. *VLSI Array Processors*. Prentice Hall, 1988.
- [KuWe86] H.T. Kung et J.A. Webb. Mapping Image Processing Operations onto a Systolic Machine. *Journal of Parallel and Distributed Computing*, 1(4):246–257, 1986.
- [LiMa89] H. Li et M. Maresca. Polymorphic-Torus Architecture for Computer Vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(3):320–330, mars 1989.
- [LoMc80] R.M. Loughhead et D.L. McCubbrey. The Cytocomputer: A Practical Pipeline Image Processor. In *Proc. 7th Annual International Symposium on Computer Architecture*, pages 271–277, juin 1980.
- [MaLi89] M. Maresca et H. Li. Connection Autonomy in SIMD Computers: A VLSI Implementation. *Journal of Parallel and Distributed Computing*, 7(2):302–320, octobre 1989.
- [MaLL88] M. Maresca, M.A. Lavin, et H. Li. Parallel Architectures for Vision. *Proceedings of the IEEE*, 76(8):970–981, août 1988.
- [MeND91] A. Mérigot, Y. Ni, et F. Devos. Architectures massivement parallèles pour la vision artificielle. *Annales des Télécommunications*, 46(1-2):78–89, janvier 1991.
- [Nick90] J.R. Nickolls. The Design of the MasPar MP-1: A Cost Effective Massively Parallel Computer . In *Proceedings of IEEE Comcon Spring 1990*, pages 25–28, IEEE, février 1990.
- [OlRe85] D.E. Oldfield et S.F. Reddaway. An Image Understanding performance study on the ICL distributed array processor. In *Proc. IEEE Workshop on Computer Architecture for Pattern Analysis and Image Database*, pages 256–264, novembre 1985.
- [Pete90] C. Peterson. 100 MOP LIW Microprocessor for Multicomputers. In *HOT Chips Symposium II*, Santa Clara University, Santa Clara, CA (USA), août 1990.

- 
- [ScWi88] L.A. Schmitt et S.S Wilson. The AIS-5000 Parallel Processor. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(3):320–330, mai 1988.
- [Sieg81] H.J Siegel, L.J. Siegel, F.C. Kemmerer, P.T. Mueller, H.E Smalley, et S.D. Smith. PASM: A Partitionable SIMD/MIMD System for Image Processing and Pattern Recognition. *IEEE Transactions on Computers*, C-30(12):934–947, décembre 1981.
- [TaHN89] I. Tamitani, H. Harasaki, et T. Nishitani. A Real-Time HDTV Signal Processor - Architecture and Implementation. In *Third International Workshop on HDTV*, août 1989.
- [Tami89] I. Tamitani, H. Harasaki, T. Nishitani, Y. Endo, M. Yamashina, et T. Enomoto. A Real-Time Video Signal Processor Suitable for Motion Picture Coding Applications. *IEEE Transactions on Circuits and Systems*, 36(10):1259–1266, octobre 1989.
- [TuRo88] L.W. Tucker et G.G. Robertson. Architecture and Applications of the Connection Machine. *IEEE Computer*, 21(8):26–38, aug 1988.
- [Uhr87] L. Uhr. *Parallel Computer Vision*. Academic Press, Boston MA, 1987.
- [VeMD86] G. Verghese, S. Mehta, et C.R. Dyer. *Image Processing Algorithms for the Pipelined Image-Processing Engine*. Research Report, Department of computer Science, University of Wisconsin-Madison, septembre 1986.



---

Unité de recherche INRIA Lorraine, Technôpole de Nancy-Brabois, Campus scientifique,  
615 rue de Jardin Botanique, BP 101, 54600 VILLERS LÈS NANCY  
Unité de recherche INRIA Rennes, IRISA, Campus universitaire de Beaulieu, 35042 RENNES Cedex  
Unité de recherche INRIA Rhône-Alpes, 46 avenue Félix Viallet, 38031 GRENOBLE Cedex 1  
Unité de recherche INRIA Rocquencourt, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex  
Unité de recherche INRIA Sophia-Antipolis, 2004 route des Lucioles, BP 93, 06902 SOPHIA-ANTIPOLIS Cedex

---

Éditeur  
INRIA, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex (France)  
ISSN 0249-6399