Briques de base pour la réalisation d'architectures parallèles specialisées

François Charot

INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Briques de base pour la réalisation d'architectures parallèles spécialisées

François Charot

N° 1977

Avril 1993

PROGRAMME 1

Architectures parallèles,
bases de données,
réseaux et systèmes distribués

ISSN 0249-6399

Briques de base pour la réalisation d'architectures parallèles spécialisées

François Charot*

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

Rapport de recherche n°1977 — Avril 1993 — 23 pages

Résumé : Les études et recherches liées au développement d'applications de traitement d'image et plus particulièrement en traitement numérique du signal vidéo requièrent des composants flexibles permettant la mise en œuvre "en vraie grandeur" des algorithmes en vue de leur validation dans un contexte temps réel.

L'objectif de ce rapport est de présenter les nouveaux circuits spécifiques programmables "briques de base" pour la réalisation d'architectures parallèles spécialisées. Les approches de type traitement de signal (DSP) de Texas Instruments et Motorola, celles orientées traitement d'image et du signal proposées par Intel (iWARP), ainsi que les circuits de traitement vidéo de Philips, NEC et ITT sont considérés.

Mots-clé : processeur spécialisé programmable , DSP, VSP

(Abstract: pto)

*charot@irisa.fr

Unité de recherche INRIA Rennes
IRISA, Campus universitaire de Beaulieu, 35042 RENNES Cedex (France)
Téléphone : (33) 99 84 71 00 – Télécopie : (33) 99 38 38 32

Building blocks for the design of specialized parallel architectures

Abstract: The study and research in the area of image processing and especially in video signal processing require flexible components that allow real-time implementation of the algorithms to be considered.

In the report, we present the new specialized programmable circuits, building blocks for the implementation of specialized parallel architectures. The dsp-based approaches from Texas Instruments and Motorola, those suited to image and signal processing proposed by Intel (iWARP), and also the video signal processsing circuits from Philips, NEC and ITT are described.

Key-words: Specialized programmable processor, DSP, VSP

Introduction

Les applications de traitement d'image et plus particulièrement les applications de traitement vidéo 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).

Les études et recherches liées au développement de telles applications complexes ont justifié le développement de composants flexibles permettant la mise en œuvre "en vraie grandeur" des algorithmes en vue de leur validation dans un contexte temps réel. Parmi les nouveaux circuits spécifiques programmables apparus sur le marché, les réalisations les plus caractéristiques incluent les approches traitement de signal (DSP) proposées par Texas Instruments ou Motorola, celles orientées traitement d'image ou de signal ou traitement numérique (processeur iWARP d'Intel), enfin, des composants appelés processeurs de traitement de signal vidéo (VSP) sont maintenant étudiés et proposées par quelques industriels (Philips, NEC et ITT).

Ces nouveaux processeurs, dont l'architecture est détaillée dans ce rapport, ont en commun les caractéristiques suivantes.

- Leurs possibilités d'entrée/sortie permettent la réalisation efficace de structures parallèles (briques de base) et la mise en œuvre de communications interprocesseurs à haut débit, et ce, sans ajout de logique externe.
- Leurs capacités de calcul (en arithmétique en virgule flottante pour certains) sont considérables.
- L'organisation de leurs architectures internes est telle que plusieurs opérations peuvent être lancées en parallèle.

1 Approche DSP

Le concept de DSP, dont le but est le traitement numérique du signal, date de 1979, époque à laquelle est apparu le premier processeur de ce type proposé par Intel. Ce concept a été adopté par les principaux fabricants de microprocesseurs. Deux familles de DSP sont apparues : le DSP programmable adaptable à toute application de type traitement de signal, et le DSP dédié à un type de traitement (filtrage, FFT, ...).

La largeur des chemins de données est de 16 ou 32 bits, voire même 24 bits, les opérations s'effectuent en virgule fixe (entiers) ou en virgule flottante. Ils comportent tous une unité arithmétique et logique, un multiplieur, des accumulateurs, un registre à barillet. La dessus, viennent se greffer de la ROM pour le programme, de la RAM pour les données et divers ports périphériques classiques (port parallèle, port série, contrôleur d'accès direct à la mémoire). Leurs architectures sont généralement de type Harvard, caractérisées par deux bus distincts, pour des accès indépendants aux données et aux instructions.

Texas Instruments est la firme possédant le catalogue le plus important avec les familles de processeurs 320C1x, 320C2x, 320C3x, et maintenant les familles 320C4x et 320C5x. La série C4x, détaillée ci-après, reprend les concepts de la série C3x, mais en développant l'aspect multiprocesseur. La série C3x se caractérise par des opérations en virgule flottante 32 bits, un temps de cycle de 60 ns (33 MFLOPS), une mémoire de données de 2 K mots, une mémoire cache d'instructions de 64 mots, deux ports série et un contrôleur d'accès direct à la mémoire. Le circuit C30, conçu en technologie CMOS 1 μm intègre 700000 transistors dont la moitié réalise la mémoire. La série C5x est une version plus rapide (temps de cycle de 50 ns et 35 ns) des 16 bits à virgule fixe (séries C10, C20).

Le catalogue d'ATT comporte des DSP en virgule flottante 32 bits (DSP32C, 25 MFLOPS à une fréquence de 50 MHz, 4 K mots de RAM, 16 K mots de ROM), et également des processeurs 16 bits (DSP16A, 40 MIPS avec un temps de cycle de 25 ns, mémoire cache de 15 mots, 2 K mots de RAM, 12 K mots de ROM).

Motorola propose la famille 5600x fonctionnant sur 24 bits (27 MHz, 1 K mots de RAM, 544 mots de ROM), et maintenant le 96002 qui reprend l'architecture du 56001, mais travaille sur 32 bits en virgule flottante et traite une instruction en 60 ns (50 MFLOPS sur certains traitements).

Les inconvénients majeurs des DSP actuels sont doubles : leurs utilisations dans un contexte multiprocesseur nécessitent des développements matériels lourds car il faut ajouter beaucoup de logique pour réaliser l'interconnexion et la commu-

nication entre processeurs, leurs possibilités d'entrée-sortie souvent limitées rendent difficile la mise en œuvre de communication à haut débit entre processeurs, leur programmation est souvent difficile (problèmes de compilation). Deux DSP échappent à ces critiques et sont décrits ici de façon plus détaillée : le Texas Instruments TMS320C40 et le Motorola DSP96002. Il s'agit de processeurs 32 bits ayant d'énormes possibilités en ce qui concerne le traitement en arithmétique en virgule flottante. De plus ces processeurs sont intéressants de part leur organisation architecturale interne et leurs possibilités de parallélisme.

1.1 Circuit Texas Instruments TMS320C40

En octobre 1990, Texas Instruments annonce le successeur du TMS320C30 : le "C40' Le C40 [Sima91] est un processeur de traitement de signal 32 bits intégrant une unité de calcul en arithmétique flottante et disposant de facilités pour la réalisation de systèmes parallèles, comme illustré par la figure 1. En effet, le C40 dispose de six ports de communication rapides pour les échanges directs entre processeurs sans ajout de logique externe. Chaque port supporte un débit de transfert bidirectionnel de 20 méga-octets/seconde.

L'unité centrale se compose d'un multiplieur et d'une unité arithmétique en virgule flottante, de 8 registres auxiliaires, de 16 registres de contrôle et de 12 registres généraux (soit au total 40 registres), et de deux unités de génération d'adresse. De nombreux bus internes assurent la liaison entre ces différents modules. Cette unité centrale peut effectuer jusqu'à huit opérations par cycle : deux accès mémoire, une opération de multiplication, une opération d'UAL, deux mises à jour d'adresse, un branchement, une mise à jour de compteur de boucle (275 MOPS avec un temps de cycle de 40 ns).

Un coprocesseur de type DMA (*Direct Memory Access*) intégré réalise les transferts rapides entre l'espace mémoire du C40 et l'extérieur. Il gère 6 canaux DMA et permet d'effectuer simultanément 6 transferts.

Le C40 a un espace d'adressage de 4 giga mots de 32 bits. Sa mémoire interne est organisée en 2 blocs RAM de 1 K mots de 32 bits et un bloc ROM de 4 K mots de 32 bits. Chaque bloc est capable de supporter 2 accès par cycle. En un cycle, l'unité centrale peut par exemple accéder à deux données dans un bloc RAM et aller chercher une instruction dans une mémoire externe en parallèle avec le chargement DMA du second bloc RAM.

L'exécution des instructions est contrôlée par cinq unités : la lecture de l'instruction en mémoire, le décodage de l'instruction et la génération d'adresse des

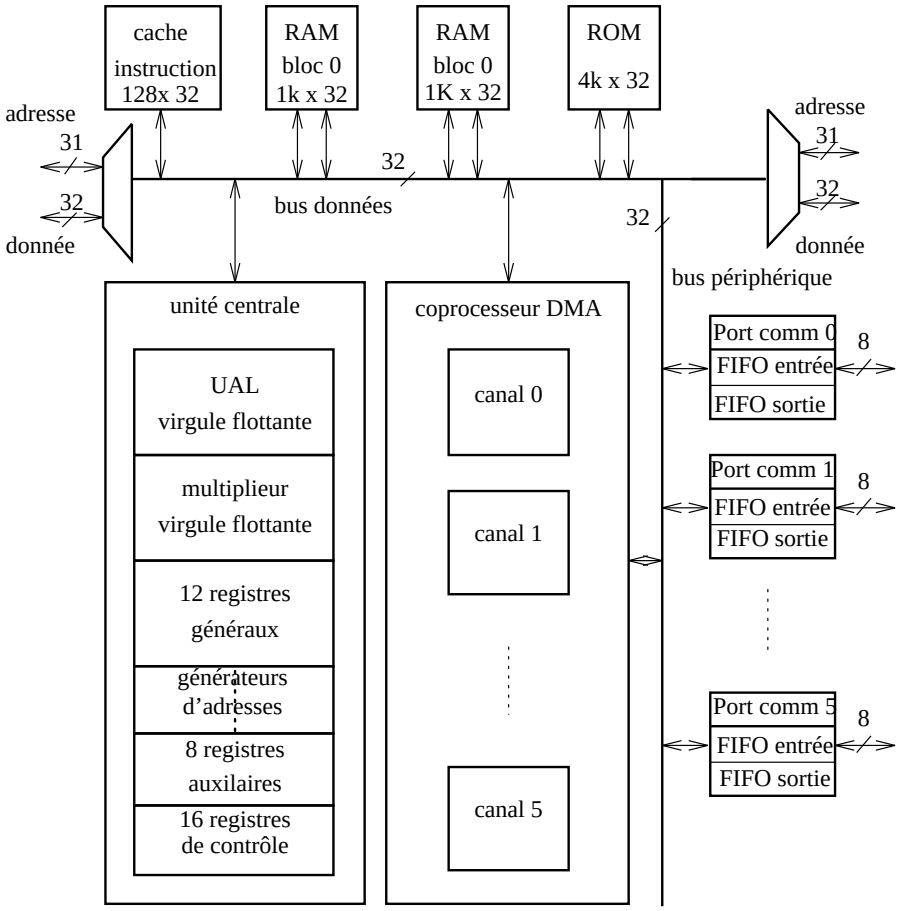

Figure 1 : Organisation du circuit TMS320C40.

opérandes, la lecture des opérandes en mémoire, l'exécution (lecture des opérandes en registres, opération et écriture du résultat en registre), l'opération DMA (lecture, écriture mémoire). Une instruction de base a 4 étages de pipeline (extraction, décode, lecture et exécution). Ce processeur n'a pas de mémoire programme interne mais un cache. Celui-ci a une capacité de 128 mots de 32 bits.

Un exemple d'instruction parallèle codée sur 32 bits :

FMPY *(AR5)++(1),*–(AR1)(IR0),R0 FADD R5,R7,R3

L'exécution de l'instruction (FMPY *(AR5)++(1),*–(AR1)(IR0),R0) multiplie la valeur lue en mémoire à l'adresse spécifiée par le registre d'adresse AR5 par la valeur lue en mémoire à l'adresse spécifiée par le registre d'adresse AR1, et stocke le résultat dans R0. (FADD R5,R7,R3) additionne R5 et R7 et stocke le résultat dans R3. En parallèle, les registres d'adresse AR5 et AR1 sont mis à jour (AR5 est incrémenté de 1, AR1 = AR1 - IR0).

1.2 Circuit Motorola DSP96002

Le circuit DSP96002 est un processeur à opérateurs supportant de l'arithmétique en virgule flottante 32 bits dont les principales composantes sont les suivantes (figure 2) :

- L'unité d'échange de données consiste en 5 bus bidirectionnels 32 bits (bus données X et Y, bus global G, bus DMA D, bus programme P). Les transferts de données entre l'UAL et les mémoires X, Y sont réalisés via les bus X, Y.

- Une UAL réalise toutes les opérations arithmétiques et logiques. Elle consiste en 10 registres généraux 96 bits qui peuvent être vus comme 30 registres 32 bits, un décaleur à barillet 32 bits, un additionneur 32 bits, un multiplieur 32 bits ainsi qu'un diviseur. Toutes les opérations sont réalisées en un cycle. Les opérandes proviennent toujours des registres généraux, un résultat est toujours mémorisé en registre. Les bus de transfert de données (X, Y, G) sont libres pendant ces opérations.

- Deux mémoires de données double port X et Y consistent en 512 mots de 32 bits de RAM et 1024 mots 32 bits de ROM chacune. Il est à noter que ces espaces mémoire sont indépendants.

- L'unité d'adressage réalise les calculs d'adresse pour l'accès des opérandes en mémoire ainsi que la mémorisation des adresses. Elle opère en parallèle avec les autres ressources et comporte 24 registres 32 bits accessibles via le bus G (8 registres d'adresse, 8 registres d'offset contenant des incréments/décréments d'adresse, 8 registres de modification spécifiant le type d'adressage), des registres d'adresse temporaires, deux unités arithmétiques : les types d'adressage peuvent être linéaires, modulo, etc. Deux adresses sont calculées à chaque cycle (pour chaque espace mémoire). La

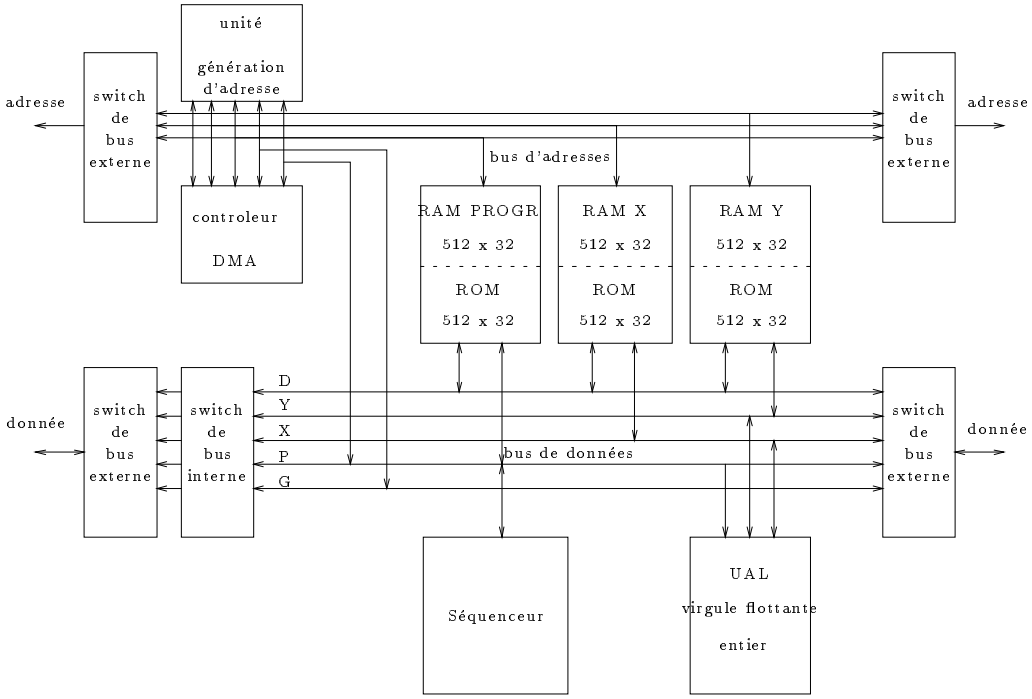

Figure 2 : Organisation du circuit DSP96002.

plupart des modes d'adressage modifie le registre adresse sélectionné selon l'approche *read-modify-write*.

- L'unité de contrôle réalise l'extraction des instructions, leur décodage, le contrôle des boucles *do* ainsi que le traitement des exceptions. La mémoire programme interne a une capacité de 1024 mots de 32 bits.

- Deux bus externes 32 bits assure l'interface avec des mémoires externes données, programmes ou des unités d'entrée/sortie.

Sa structure à bus multiple autorise jusqu'à 3 transferts de données dans une instruction par l'intermédiaire des bus X, Y et G. Aussi, une instruction 32 bits permet de spécifier une multiplication, une addition, un transfert de la mémoire X vers un registre avec post incrémentation de l'adresse mémoire ainsi qu'un

transfert de la mémoire Y vers un registre avec post incrémentation de l'adresse mémoire.

Un exemple d'instruction parallèle codée sur 32 bits :

FMPY D4,D5,D0 FADD D0,D1 X:(R0)+,D4 Y:(R4)+,D5

L'exécution de l'instruction (FMPY D4,D5,D0) multiplie la valeur du registre D4 par la valeur du registre D5, stocke le résultat dans D0. (FADD D0,D1) additionne D0 et D1 et stocke le résultat dans D0 (D0, D1, D4, D5 sont des registres interne à l'UAL). En parallèle, D4 et D5 sont mis à jour avec des valeurs provenant des mémoires de données, les registres d'adresse R0 et R4 sont incrémentés de 1 (R0, R4 sont des registres internes à l'unité de génération d'adresse).

L'architecture interne du 5600x est très voisine de celle du 96002. Ses principales différences concernent les entrées/sorties (pas de contrôleur DMA, ...), le format de données 24 bits entier (figure 3).

Il est à noter qu'un noyau DSP 16 bits a été développé à partir de la version 24 bits 56001 [LeCG90]. Ses performances sont tout à fait intéressantes : temps de cycle de 25 ns à 80 MHz. Ce noyau se compose d'une UAL, d'une unité de génération d'adresse et d'un contrôleur reliés par l'intermédiaire de 3 bus de données et 3 bus d'adresse. Ce noyau a la capacité de calculer à chaque instruction 3 adresses (2 pour l'accès en mémoire de 2 données et une pour le séquencement). Dans une instruction sont réalisés : 2 accès mémoire (RAM interne de 2 K mots 16 bits double port), 2 calcul d'adresse, 1 opération UAL de type multiplication/accumulation et l'opération de séquencement, soit au total 7 opérations. Le DSP56116 est organisé autour de ce noyau, sa mémoire interne se compose de deux RAM de 2 K mots de 16 bits (donnés et instructions). Son espace mémoire est limité à 64 K mots 16 bits.

Un exemple d'instruction parallèle codée sur 16 bits :

MAC X1,Y1,A X:(R2)+,Y1 X:(R3)+,X1

L'exécution de l'instruction (MAC X1,Y1,A) multiplie la valeur du registre 16 bits X1 par la valeur du registre 16 bits Y1 (X1 et X2 sont deux registres internes à l'UAL), ajoute le produit au registre accumulateur A (registre interne à l'UAL) et stocke le résultat dans l'accumulateur A. En parallèle, X1 et Y1 sont mis à jour avec des valeurs provenant de la mémoire de données, les registres d'adresse

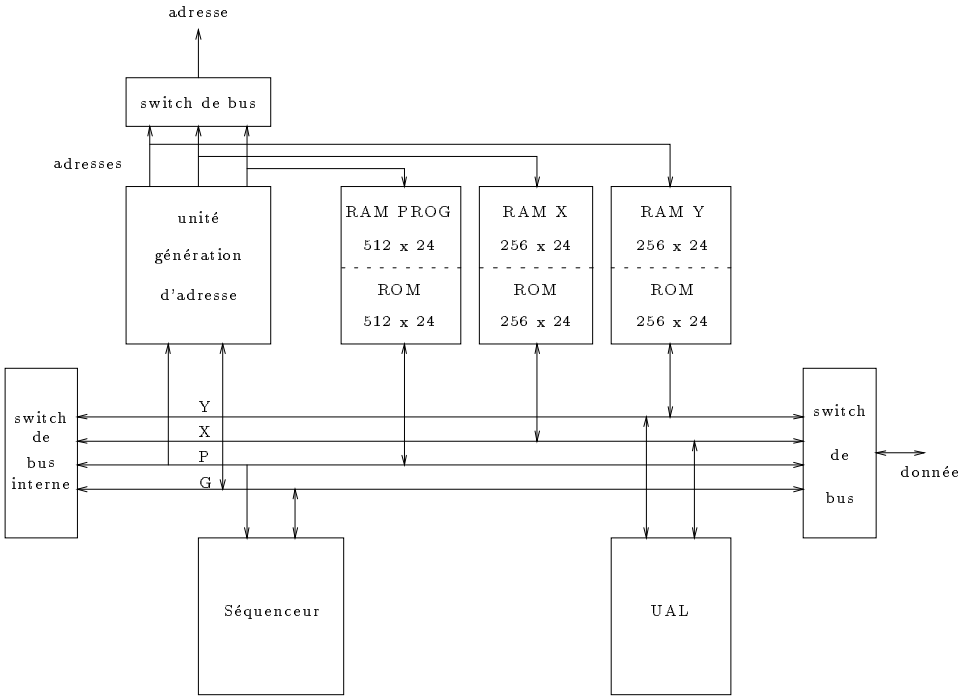

Figure 3 : Organisation du circuit DSP56000.

R2 et R3 sont incrémentés de 1 (R2, R3 registres internes à l'unité de génération d'adresse).

1.3 Comparaison de ces deux approches

Les principales différences entre ces deux DSP concernent :

- L'organisation mémoire. Les deux blocs du Texas Instruments font partie du même espace d'adressage. Dans le Motorola, il s'agit de deux espaces d'adressage distincts, ce qui rend plus difficile la tâche des compilateurs.
- Il n'y a pas de cache d'instructions dans le 96002 mais une mémoire de programme de 1024 mots de 32 bits.

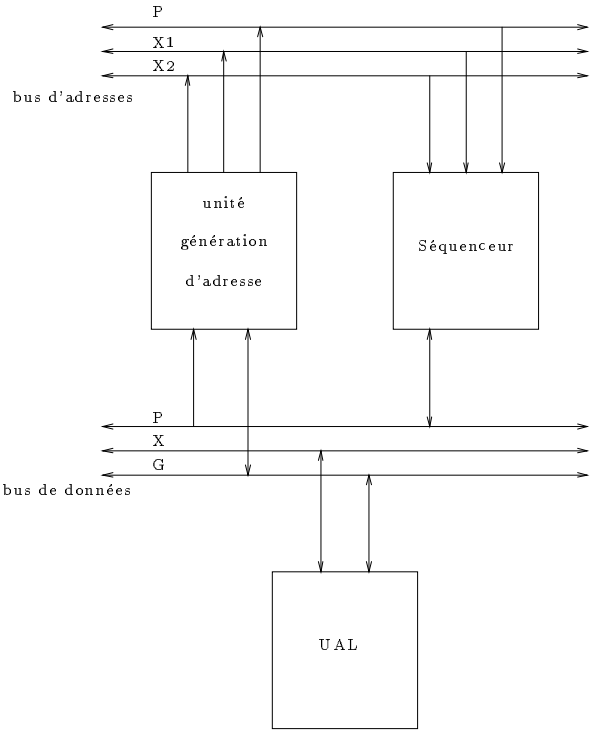

Figure 4 : Organisation du noyau DSP motorola.

- Toutes les instructions arithmétiques du 96002 ont comme opérandes des registres alors que dans le Texas Instruments, les opérandes peuvent directement provenir des mémoires.
- Les entrées/sorties. En effet, le Motorola ne dispose pas des ports de communications permettant les interconnexions directes avec d'autres DSP.

2 Circuit iWARP

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[Anna87]. Le iWARP utilise l'ap-

proche VLIW, c'est à dire que des instructions longues spécifient explicitement l'ensemble des opérations s'exécutant dans une même instruction [CGLT89]. Le format d'instruction permet de spécifier jusqu'à neuf opérations qui s'exécuteront en parallèle.

Une cellule iWARP consiste en un circuit iWARP et sa mémoire locale. Des systèmes parallèles "sur mesure" (topologie variée, taille du réseau adaptée à la complexité du problème) peuvent être aisément construits en reliant simplement les ports d'entrées/sorties des cellules iWARP.

Le circuit comporte 650 000 transistors, il atteint une puissance de 20 MFLOPS à une fréquence de 20 MHz (les transferts de données entre processeurs sont réalisés à 40 MHz).

2.1 Architecture du processeur

Le composant iWARP, dont l'organisation est donnée figure 5, consiste en trois unités autonomes.

- 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 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).

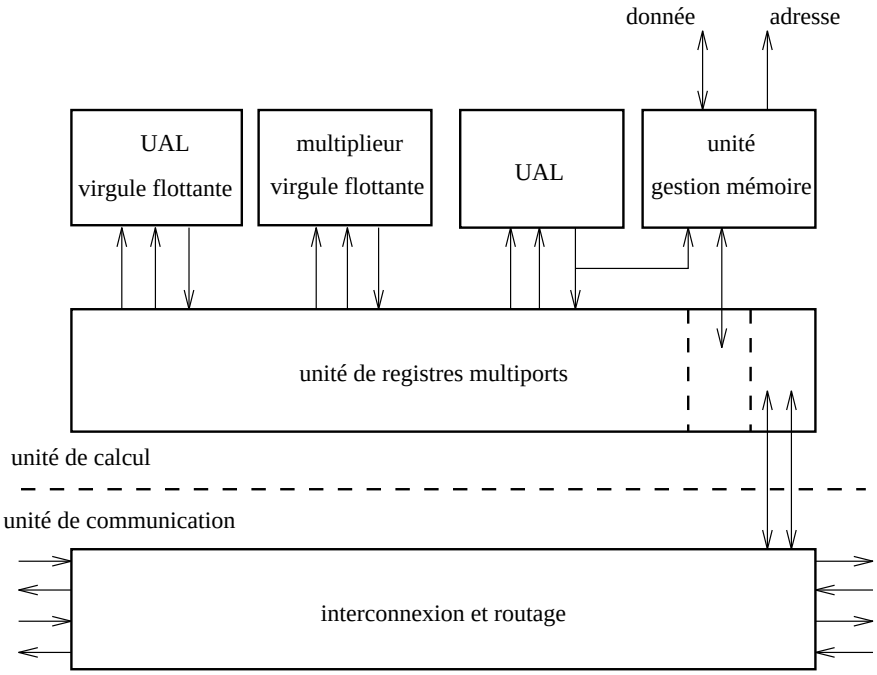

Figure 5 : Organisation du circuit iWARP.

L'unité de registres comporte 6 ports standard : deux ports vers chacune des unités arithmétiques et trois ports dédiés ou ports spécialisés assurant le lien entre des interfaces externes et des emplacements particuliers dans l'unité de registres. Ces ports spécialisés permettent le transfert de données entre la mémoire ou l'unité de communication et l'unité de registres et ce indépendamment du fonctionnement des autres unités fonctionnelles du processeur. La motivation de ces ports spécialisés est double : besoin de réduire la largeur des instructions et le nombre de ports dans le bloc de registres.

2.2 Contrôle

Le contrôle du processeur est réalisé via des instructions longues spécifiant explicitement le parallélisme entre les différentes unités fonctionnelles (jusqu'à neuf opérations peuvent être spécifiées dans une instruction). Il y a deux formats d'instructions : des instructions 96 bits (3 mots 32 bits) pour un parallélisme maximal, et des instructions courtes (un mot 32 bits). Une instruction courte contrôle une seule unité fonctionnelle (transfert de données, opération arithmétique, opération entière, branchement, ...). Il s'agit d'instructions à un ou deux opérandes registres codés sur 9 bits ou d'instructions à 3 opérandes registres codés sur 7 bits. L'instruction longue est horizontale par nature puisqu'elle permet le parallélisme entre les différentes unités fonctionnelles. Il en existe une seule, appelée (*Calcule & accède*) qui peut se décomposer en opération sur l'UAL flottante, opération sur le multiplieur et deux accès mémoire ou opération sur l'UAL entière. Des registres spécialisés de l'unité de registres servent de registres de données implicites, c'est à dire que la donnée lue en mémoire lors de l'exécution d'une telle instruction y est stockée.

Le jeu d'instructions comporte 32 instructions se répartissant en 7 classes : transfert de données, gestion de pile, opération entière/logique, opération flottante, branchement, appel/retour de sous-programme, contrôle de ressources (chemins de communication, ...). Des mécanismes matériels permettent l'exécution efficace des boucles (contrôle du nombre d'itération de la boucle, branchement, ...). Chaque instruction comporte un champ dit de terminaison (1 bit). Lorsque ce champ est positionné, il y a mise à jour du compteur d'itération, contrôle de la condition et branchement à l'adresse de début de boucle. Ces opérations sont réalisées en 2 cycles machine, temps nécessaire à l'exécution de la plupart des instructions arithmétiques. Aussi une instruction de type (*Calcule & accède*) itérée s'exécute en 2 cycles machine et ce sans pénalité.

Chaque instruction a un temps d'exécution nominal. L'exécution de certaines instructions diffère selon l'état courant de la machine, ce qui peut nécessiter un temps additionnel pour la terminaison de certaines instructions. En simple précision, une instruction arithmétique en virgule flottante s'exécute en 2 cycles machine de 50 ns, il n'y a pas de pipeline. L'instruction (*Calcule & accède*) s'exécute également en 2 cycles, les deux unités en arithmétique flottante effectuant leur opération en deux cycles, l'UAL entière réalisant un calcul d'adresse sur chacun des cycles de 50 ns. Il s'agit dans ce cas d'une utilisation implicite de l'UAL entière sous le contrôle d'une instruction à l'origine prévue pour une autre unité

fonctionnelle. En fait l'UAL entière réalise la génération d'adresse pour les opérations d'accès mémoire, le calcul des adresse de branchement, les opérations de pile.

La boucle suivante :

```
for i := 0 to n-1 do {
        a[2*i] = c + b[i] * d;
}
```

peut être codée en une instruction 96 bits.

FMPY R0, R8, R10 FADD R1, R10, R9 MEM:(R3)+,R0 R9,MEM:(R4)+

L'exécution de l'instruction (FMPY R0, R8, R10) multiplie les valeurs des registres R0 et R8 et stocke le résultat dans R10. (FADD R1, R10, R9) additionne les valeurs des registres R1 et R10 et stocke le résultat dans R9. En parallèle, une valeur lue en mémoire à l'adresse spécifiée par le registre R3 est stockée dans R0, la valeur du registre R9 est écrite en mémoire à l'adresse pointée par R4. Les adresses sont mises à jour. Sur un processeur comme le DSP96002, une telle boucle ne peut être codée sur une instruction, et s'exécute en 2 cycles (limitation de bande passante registres).

2.3 Communications avec l'extérieur

La mémoire locale est externe au processeur (espace de 16 méga-mots de 32 bits). Cet espace est à la fois partagé par les données, les instructions et la pile. L'espace instruction est accédé automatiquement sous le contrôle de l'unité de séquencement et par les instructions de contrôle telles les branchements, les appels de sous-programme, ... 2 K mots de l'espace instruction sont localisés dans une ROM interne (noyau de code pour les communications entre cellules, fonctions systèmes de base, ...). Le reste de l'espace instruction réside en mémoire locale et est exécuté à travers un cache instruction interne au processeur de 256 mots. L'espace des données est accédé par les instructions d'accès mémoire et l'instruction *Calcule & accède*. Seul des mots (32 bits) ou double mots sont transférés entre la mémoire locale et le processeur.

L'unité de communication a 4 ports d'entrée, 4 ports de sortie, supportant chacun une bande passante de 40 méga-octets/s. Chaque port possède les mécanismes

de contrôle de flux (5 bits de statut). Un port de sortie peut être directement connecté à l'un quelconque des ports d'entrée d'un autre processeur. Des bus logiques multiples peuvent être multiplexés sur chaque bus physique, jusqu'à 20 chemins peuvent être ainsi gérés simultanément (contrôle et mémorisation). Les modes de communication systolique et par messages sont supportés.

2.4 Machines à base de iWARP

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 2×2 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 2×16 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 a une station de travail SUN grâce à une ou plusieurs cartes d'interface.

2.5 Programmation

Le logiciel fournit avec les architectures iWARP est composé d'un environnement de développement sur station de travail SUN ainsi qu'un noyau d'exécution fonctionnant à la fois sur SUN et chaque processeur iWARP.

L'environnement de développement est constitué de "*cross-compilateur*" C et Fortran. Le compilateur C génère du code optimisé pour le iWARP. Des extensions à ces langages permettent la communication entre processeurs selon différents paradigmes.

Intel propose également le langage APPLY et ASSIGN :

- Le langage APPLY est plus particulièrement dédié à la programmation d'applications de traitement d'images. Une librairie (Weblib) pour le traitement de bas niveau (traitement sur les pixels) est fournie.

- Le langage ASSIGN est dédié à la programmation d'applications de traitement du signal.

3 Circuit VISP NEC

Dans le cadre de ses études de systèmes multiprocesseurs pour le codage d'image, NEC a développé un processeur VLSI spécifique à architecture parallèle intégrant de nombreuses fonctions nécessaires au traitement numérique du signal vidéo.

Ce processeur [Nish89] 16 bits se compose des unités suivantes : deux unités de génération d'adresse (UGA) , une unité de gestion de temps (GT), une unité de traitement (UT). L'unité de traitement est organisée autour de 3 bus (A, B pour le transfert d'opérandes vers l'unité arithmétique, C pour le transfert des résultats émis par l'unité de traitement). L'unité de traitement consiste en une unité arithmétique pipeline reconfigurable (UAP), une unité de contrôle (UC) (séquencement et RAM programme de 512 mots de 32 bits) et une unité de mémorisation (UM) composée de 2 RAM de 128 mots de 16 bits.

L'unité arithmétique a 7 étages de pipeline pour réaliser efficacement les calculs basés sur les normes L1, L2 et ce en pipeline. Parmi les étages du pipeline, on note l'étage de calcul (UAL et multiplieur au choix pouvant réaliser des opérations de valeur absolue de différence), l'étage d'accumulation de sommes partielles, l'étage de calcul de minimum/maximum sur une séquence de distances avec mémorisation de l'ordre.

Les unités de mémorisation externes d'entrée (ME1, ME2) sont accédées par les unités de génération d'adresse du processeur et reliées aux ports d'entrée du processeur. En sortie, des FIFOS sont utilisées (FS1, FS2).

Le circuit, conçu en technologie CMOS 1,2 μm intègre 220000 transistors et fonctionne à 40 MHz.

4 Circuit VSP Philips (Video Signal Processor)

Les contraintes inhérentes aux applications vidéo temps réel (haute fréquence d'échantillonnage, multitude de types de signaux à traiter, large éventail de traitements allant du très régulier à l'irrégulier) ont conduit à la notion de processeur de traitement vidéo (Video Signal Processor VSP).

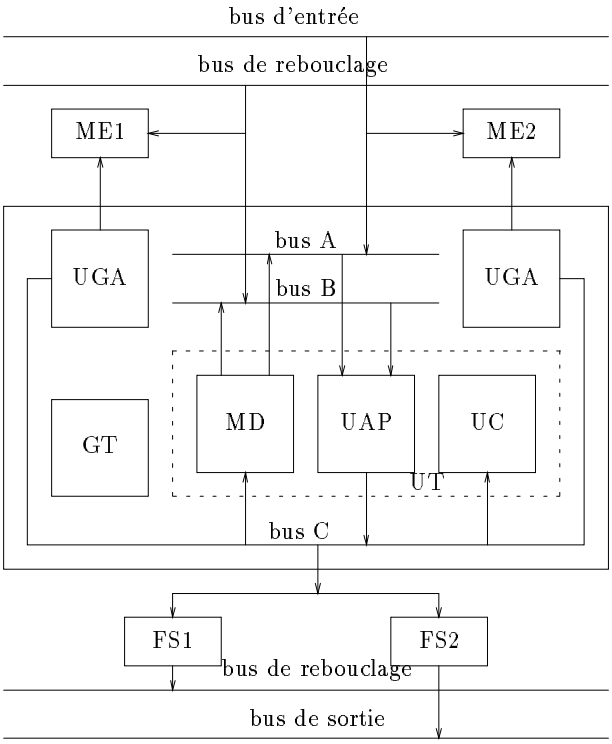

Figure 6 : Organisation du circuit VISP.

Philips a développé un processeur de traitement vidéo [Roer89] [Slui89] capable de supporter les différents types de traitements à l'exception des mémoires de trames et des démultiplexages (lorsque la fréquence d'échantillonnage excède 27 Mhz).

4.1 Architecture du processeur

Le VSP est un bloc de base modulaire pour la réalisation de systèmes dédiés "taillés sur mesure". Le VSP proposé a au plus 3 canaux d'entrée et 3 canaux de sortie (sur une largeur de 12 bits) et consiste en trois "modules" de traitement et une unité d'initialisation. Le programme d'application est chargé à partir d'une entrée série sous le contrôle de l'unité d'initialisation. Un programme peut être

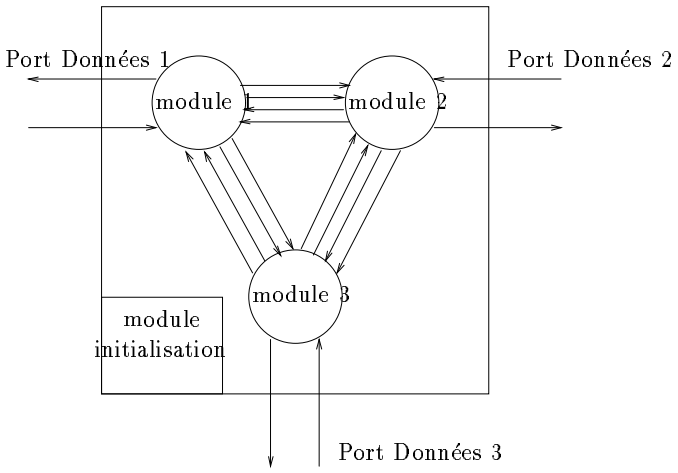

Figure 7 : Organisation du circuit VSP.

envoyé à plusieurs VSP en parallèle, chaque VSP identifie la partie de programme le concernant (une adresse marque le code propre à chaque VSP. Les "modules" de traitement sont reliés entre eux par l'intermédiaire de 4 canaux 12 bits (2 canaux d'entrée, 2 canaux de sortie). Dans une technologie 1,6 μm, il est possible d'intégrer 1 module dans un circuit, en technologie 1 μm, 3 modules identiques peuvent être intégrés sur la même surface de silicium.

Un "module" est un ensemble de 10 unités programmables (PU). Les unités sont toutes synchronisées par la même horloge (27 MHz). Chaque PU a son propre programme. Un programme consiste en au maximum 16 instructions, celles-ci sont répétées cycliquement à raison d'une instruction par cycle. Pour communiquer avec des modules identiques, celui-ci dispose de 5 ports d'entrée et 5 ports de sortie.

Trois PUs consistent en unités arithmétique et logique (UAL) 12 bits, chaque UAL a 3 entrées, 2 entrées opérandes et une entrée spécifique à certaines opérations. Deux PUs sont des éléments de mémorisation (UM) pour la stockage d'une partie de ligne d'image ou la mise en œuvre de délais et de tables de look-up. Chaque mémoire consiste en 512 mots de 12 bits et a deux entrées, une entrée adresse et une entrée données, l'adresse de la mémoire est la somme d'une adresse externe et

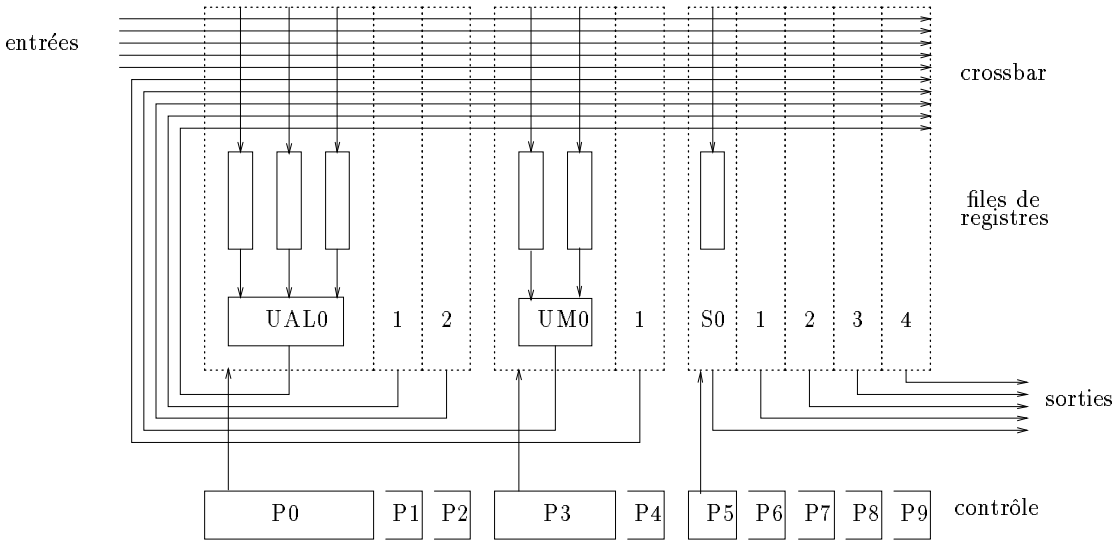

Figure 8 : Architecture du module de VSP.

d'un paramètre issu de l'instruction. Les cinq autres PUs incorporent des tampons de sortie (sortie vers les "modules" voisins ou les ports externes du VSP.

Un réseau crossbar réalise les interconnexions entre les PUs, ainsi qu'entre les entrées du VSP et les PUs, ce réseau est configuré dynamiquement à chaque cycle. A chaque entrée de PUs est associée une file de 32 registres pipelines, permettant ainsi la mise en œuvre de délais programmables. Le fonctionnement de cette unité spécialisée est le suivant : à chaque cycle d'horloge, l'un quelconque des registres de l'unité est accessible en lecture, seul le registre extrême de l'unité (registre 0) est accessible en écriture. A chaque écriture dans le registre 0, il y a décalage des données sur toute l'unité de registres[1] Les PUs opèrent en parallèle et sont pipelinés pour des performances maximales.

Le VSP a été réalisé en technologie CMOS 1,6 μm (120mm^2, 206000 transistors, 208 broches, dont 120 pour les entrées/sorties, bande passante : 3,2 Gbits/s).

[1]Ces registres reliés entre eux peuvent être vus comme un grand registre à décalage.

4.2 Programmation

Des outils pour faciliter la programmation ont été développés. L'utilisateur intervient interactivement au niveau graphique, tous les aspects internes dépendants de l'architecture sont cachés au programmeur (outils de partitionnement/placement interactifs) [Dijk89].

La stratégie de programmation d'une application consiste en un certain nombre d'étapes.

- Construction d'un graphe flot de données ou chaque nœud spécifie une opération élémentaire, sélectionnée dans le jeu d'instruction du processeur. L'outil est capable de prendre en compte différentes fréquence d'échantillonnage du signal.
- Estimation du nombre de modules nécessaire et construction d'un réseau d'interconnexion entre les modules.
- Placement du graphe sur l'architecture (choix du module, du processeur, du temps d'exécution. Cette phase consiste en un partitionnement du graphe en sous-graphes, chaque sous-graphe est exécuté sur un module (prise en compte des capacités de traitement du module, de ses possibilités de communication). Le partitionnement est suivi d'un ordonnancement (prise en compte des dépendances de données, du pipeline, ...).
- Enfin, le code de l'application est produit.

5 Composant Datawave ITT

ITT a annoncé au cours du premier semestre 1990 un composant multiprocesseur à traitement parallèle, piloté par le flot des données et de type MIMD [ScCH90]. Ce composant de nom *DataWave* est réalisé en technologie CMOS 0,8μm, il contient 1,2 million de transistors sur une surface de 150 mm^2 et a une performance de 4 GOPS.

DataWave est un processeur composé de 16 cellules de calcul. Chaque cellule est un processeur de type RISC. Conçu pour fonctionner avec une fréquence de 125 MHz, chaque cellule atteint une performance de 250 MOPS. Une structure de communication doit permettre d'exploiter la puissance de calcul : la fréquence de transfert maximale annoncée se situe à 750 Mo/s.

Les cellules sont arrangées selon une matrice 4x4. Chaque cellule peut communiquer avec ses voisines dans les quatre directions (nord, sud, est et ouest) grâce à deux bus unidirectionnels de 12 bits chacun. Tous les chemins de données et l'architecture des structures de calcul sont fondés sur des largeurs de mots de 12 bits. Cette largeur de mots convient bien aux applications en traitement vidéo ou les résultats sortant des convertisseurs analogique/numérique sont en général sur 8 ou 9 bits.

Une cellule est composée d'un multiplieur/accumulateur, d'une unité arithmétique et logique, d'un registre à décalage, d'une pile de 16 registres à 3 ports d'accès et d'une mémoire de programme de 64 mots de commande de 48 bits. Le tout est entouré de 3 bus en anneaux (figure 9). La mémoire de programme est connectée à un bus série permettant de charger des informations depuis un environnement externe. Les opérandes de calcul sont fournis par deux bus internes de 12 bits, qui reçoivent les informations depuis la pile de registres, des bus en anneaux et des constantes présentes dans le programme. Chaque unité de calcul possède 6 niveaux de pipeline. Deux opérations peuvent être réalisées pendant chaque cycle d'horloge (par exemple une addition et une multiplication). Avec une fréquence de 125 MHz il est possible de démarrer une nouvelle opération à chaque cycle d'horloge : un cycle d'horloge dure 8ns. En outre, grâce à un rebouclage en interne, le résultat d'une opération effectuée dans le multiplieur/accumulateur ou dans l'unité arithmétique et logique peut être exploité lors de l'opération suivante.

A une telle fréquence de fonctionnement, des communications synchrones sont difficiles à mettre en œuvre, et ce même si toutes les cellules sont alimentées par le même signal d'horloge, des irrégularités dans les lignes d'horloge peuvent conduire à des dérives de celle-ci. Les communications asynchrones permettent une mise en œuvre simple de communications inter-processeurs à l'extérieur du circuit et ce même avec des processeurs fonctionnant à des fréquences d'horloge différentes. Des mémoires FIFOS en entrée et en sortie de chaque cellule permettent ainsi de réduire les cycles d'attente à un minimum.

6 Conclusion

L'émergence de nouvelles applications, en particulier dans le domaine de l'image et du traitement vidéo provoque l'apparition de nouveaux circuits, briques de base pour la réalisation de structures parallèles de configurations variées. Il s'avère néanmoins qu'à l'heure actuelle, il n'existe sur le marché aucun composant pro-

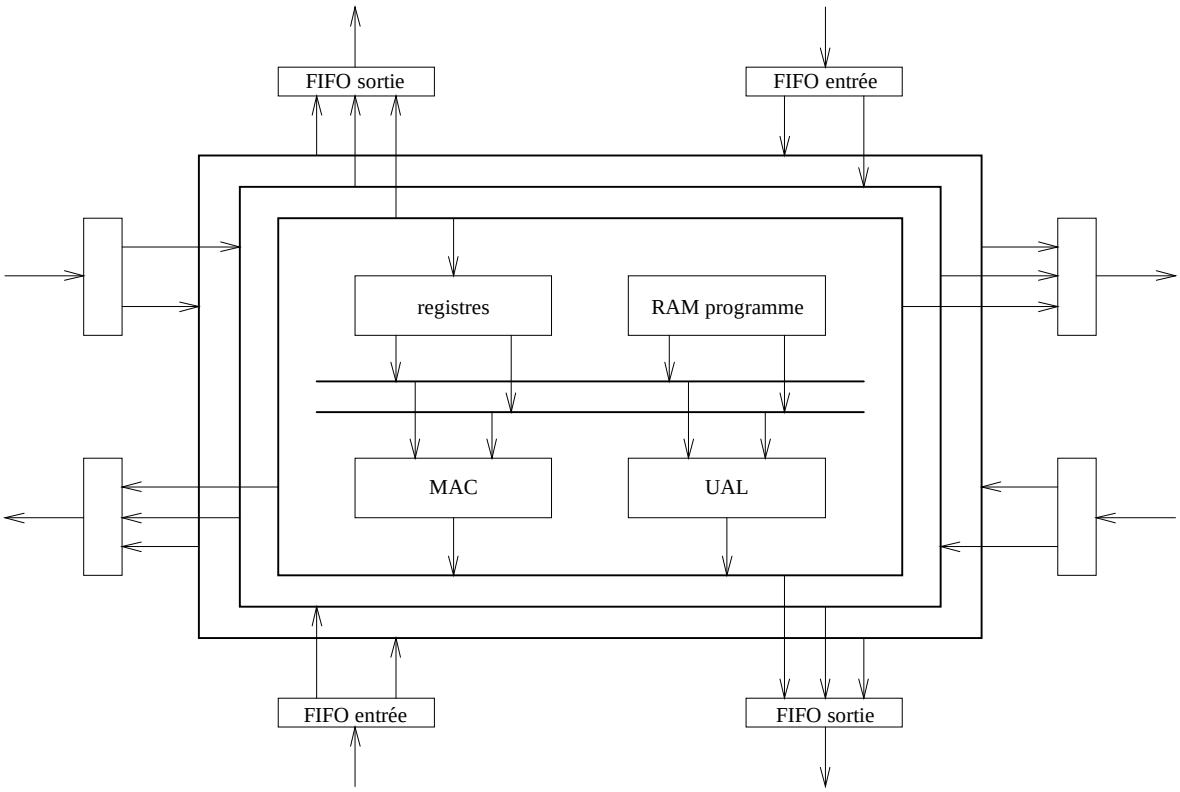

Figure 9 : Architecture de la cellule Datawave.

grammable permettant de traiter des applications complexes et ce en temps-réel. Quelques réalisations intéressantes ont cependant été proposées à ce jour mais les composants restent bien souvent inaccessibles, soit en raison du caractère expérimental de ces travaux, soit parce que l'exploitation industrielle des résultats de recherche exclut la commercialisation du composant seul.

Parmi les approches jugées les plus intéressantes, on note les approches de type traitement de signal. Les processeurs DSP actuels ont d'énormes possibilités en ce qui concerne les traitements en arithmétique flottante et disposent même maintenant de mécanismes permettant leur utilisation en réseau (TMS320C40, Motorola 96002).

La machine iWARP réalisée par Intel a des possibilités remarquables puisqu'un processeur a la puissance d'un DSP couplée avec des primitives facilitant les échanges de données entre processeurs. La configuration maximale consiste en 1024 processeurs. La taille mémoire de l'ensemble atteint 1536Mo et la puissance de crête est de 20480 MFLOPS.

Enfin des circuits spécialisés pour le traitement vidéo, issus de travaux autour de la TVHD sont maintenant proposés, circuits (VSP de Philips et *Datawave* de ITT. Il s'avère que ces circuits ne seront réellement exploitables qu'à la condition qu'un environnement de programmation efficace leur soit associé.

References

[Anna87] M. Annaratone, E. Arnould, T. Gross, H.T. Kung, M. Lam, O. Menzilcioglu, et J. Webb. The Warp Computer: Architecture, Implementation and Performance. *IEEE Transactions on Computers*, C-36(12):1523–1538, décembre 1987.

[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.

[CGLT89] R. Cohn, T. Gross, M. Lam, et P.S. Tseng. Architecture and Compiler Tradeoffs for a Long Instruction Word Microprocessor. In *Proceedings of The Third International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, pages 2–14, Boston MA (USA), avril 1989.

[Dijk89] H. Dijkstra, G. Essink, A.J. Hafkamp, H. Hengst, C.M. Huizer, A.H. Roermund, R.J Sluijter, et P.J Snijder. A General-Purpose Video Signal

Processor: Architecture and Programming. In *Proc. of the 1989 IEEE Int. conf. on Computer Design:VLSI in Computers and Processors*, pages 74–76, octobre 1989.

[LeCG90] J.C. Lee, E. Cheval, et J. Gergen. The MOTOROLA 16-bit DSP ASIC core. In *Proc. of the IEEE Int. conf. on Acoustics, Speech and Signal Processing (ICASSP)*, pages 973–976, Albuqerque NM (USA), avril 1990.

[Nish89] T. Nishitan, I. Tamitani, H. Harasaki, Y. Endo, T. Kanou, et K. Kikuchi. parallel Video Signal Processor Configuration based on Overlap-Save Technique and its LSI Processor Element: VISP. *Journal of VLSI Signal Processing*, 1(1):25–34, février 1989.

[Roer89] A.H. Roermund, P.J Snijder, H. Dijkstra, C.G. Hemeryck, C.M. Huizer, J.M. Schmitz, et R.J Sluijter. A General-Purpose Programmable Video Signal Processor. *IEEE Transactions on Consumer Electronics*, 35(3):249–258, août 1989.

[ScCH90] U. Schmidt, K. Caesar, et T. Himmel. Data-Driven Array Processor for Video Signal Processor. *IEEE Transactions on Consumer Electronics*, 36(3):327–333, août 1990.

[Sima91] R. Simar. The TMS320C40: A DSP for Parallel Processing. In *Proc. of the IEEE Int. conf. on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1089–1092, Toronto (Canada), mai 1991.

[Slui89] R.J Sluijter, P.J Snijder, H. Dijkstra, C.M. Huizer, et A.H. Roermund. A Programmable Video Signal Processor. In *Proc. of the IEEE Int. conf. on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2476–2479, mai 1989.

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