

# Approche de conception du processeur vidéo programmable MOVIE

## Design Approach of the MOVIE Programmable Video Processor

par François CHAROT, Gwendal LE FOL, Charles WAGNER

Irisa  
Campus de Beaulieu, 35042 Rennes Cedex  
email : charot@irisa.fr

### *résumé et mots clés*

Cet article décrit l'approche de conception du circuit MOVIE, brique de base pour la réalisation de simulateurs temps réel d'algorithmes de compression vidéo. Le circuit MOVIE est une petite machine systolique composée d'un processeur d'entrée/sortie et d'un réseau linéaire de processeurs de calcul, il inclut les mécanismes adaptés à l'acquisition et à la restitution de la vidéo ainsi que des facilités pour la réalisation de réseaux de calcul de configurations différentes. Les outils logiciels, conçus conjointement avec l'architecture, permettent une programmation en langage évolué et une génération de code efficace. L'architecture du circuit MOVIE est entièrement spécifiée à l'aide du langage de description de matériel VHDL, sa synthèse est réalisée avec l'outil COMPASS.

Circuit VLSI, architecture systolique, architecture SIMD, compression d'image, traitement vidéo

### *abstract and key words*

This article describes the design approach of the MOVIE circuit, building block for the development of software-only solutions for real time video processing applications. The MOVIE circuit can be seen as a small linear systolic-like array of computation processors, connected at each end to an I/O processor. Externally, the chip is provided with four bidirectional data ports and three bidirectional data video port, allowing specialized parallel and programmable architectures of various configurations to be realized. Software tools, designed simultaneously with the definition of the architecture, especially as regards the code optimization aspects, allows high level programming and efficient code generation. The MOVIE architecture has been entirely specified using the VHDL hardware description language. Its synthesis is performed under the Compass tools.

VLSI circuit, systolic architecture, SIMD architecture, video compression, video processing

## 1. introduction

Les applications de traitement vidéo et plus particulièrement les applications de compression vidéo deviennent de plus en plus complexes. Par exemple, les applications de codage vidéo MPEG 1-2 requièrent une puissance de traitement de quelques dizaines de GOPS (giga opérations par seconde). De plus, la diversité des algorithmes ainsi que leurs nombreux paramètres laissent au concepteur une grande liberté lors de l'implémentation. L'exploration et la comparaison de différentes solutions algorithmiques requièrent l'utilisation d'architectures programmables. Le

concept de « boîte à outils » dans MPEG-2 et les propositions en vue de la future norme MPEG-4 ne font qu'accentuer ce besoin.

La puissance de calcul requise pour simuler des applications MPEG est très largement supérieure à celle disponible aujourd'hui sur les microprocesseurs à usage général. Les quelques études d'implantation d'algorithmes de compression vidéo sur des machines parallèles à usage général ont mis en évidence leur inadéquation en ce qui concerne les entrées/sorties. En effet, même si leur débit brut d'entrée/sortie est potentiellement supérieur à celui nécessaire pour traiter la vidéo en temps réel, les entrées/sorties pénalisent les calculs car elles utilisent les réseaux de communication inter-processeurs.

La puissance de calcul requise peut être obtenue à un coût raisonnable par l'utilisation de matériels très spécialisés et de circuits VLSI dédiés. De telles solutions, peu programmables, restreignent de façon considérable les possibilités d'adaptation et de modification des algorithmes et de leurs nombreux paramètres. Même si de telles solutions satisfont les exigences des applications, leur manque de flexibilité ne les rend pas adapté à l'évaluation, au test et à la comparaison des nombreux algorithmes de compression.

En réponse à ces besoins, l'approche MOVIE vise à fournir une approche complètement logicielle dans le contexte du traitement vidéo temps réel. L'architecture MOVIE consiste en un réseau linéaire SIMD dédié au traitement vidéo. Son bloc de base associe un réseau de processeurs de calcul  $PE_c$  et un processeur d'entrée/sortie  $PE_{es}$ . Ce bloc de base ou module peut être vu comme un petit réseau systolique de processeurs de calcul relié aux extrémités au processeur d'entrée/sortie. Les modules, organisés en réseau linéaire, communiquent avec leur voisins par l'intermédiaire de deux ports de communication. Ceux-ci comportent également trois ports vidéo. Les éléments matériels, aisément interconnectables, permettent un assemblage rapide d'architectures parallèles semi-spécialisées. Leur architecture est conçue dans le souci du respect de l'équilibre entre les calculs et les communications en vue d'exécuter des algorithmes à parallélisme à grain fin. Les applications sont décrites en langage de haut niveau, le compilateur engendre du code pour les différents types de processeurs (processeurs de calcul et processeur d'entrée/sortie).

La section 2 de ce papier discute, après une présentation de l'état de l'art du domaine, les objectifs de ces travaux, le contexte et les hypothèses qui ont motivé les choix d'architecture. L'architecture du circuit MOVIE est détaillée dans la section 3. Les outils logiciels, base de l'environnement de programmation de MOVIE, sont décrits dans la section 4. La mise en œuvre d'une application MPEG-2 est expliquée dans la section 5. Enfin, nous présentons, dans la section 6, l'approche de conception qui vise une synthèse complète du circuit MOVIE.

## 2. approches matérielles pour le codage vidéo

Le traitement temps réel d'un schéma de codage comme MPEG-1 suppose une puissance de traitement d'environ 30 GOPS si une approche par corrélation de blocs à recherche exhaustive est employée pour l'estimation de mouvement. Il y a de plus un réel besoin de programmabilité pour satisfaire les exigences des applications. Un codeur MPEG complètement programmable exige en fait la puissance de traitement d'un supercalculateur.

Les premières implémentations matérielles ont été basées sur des approches « jeux de composants », où chaque bloc fonctionnel

de l'application (transformée en cosinus discrète, estimation de mouvement, codage à longueur variable) était mis en œuvre sur un circuit. Les progrès de la technologie VLSI ont permis, au cours des dernières années, des implantations mono-circuit de codeurs vidéo.

Les implantations récentes des standards de compression vidéo peuvent être classées en deux catégories : les circuits spécialisés non programmables et les processeurs de traitement de signal vidéo (VSP) [21, 4]. Les circuits spécialisés réalisent un standard particulier de compression selon une approche pipeline des traitements, les fonctions réalisées étant câblées. Les processeurs VSP sont des extensions des processeurs de traitement de signal à usage général DSP. Ce sont des processeurs programmables, construits à partir de cœurs de processeurs RISC ou DSP et d'unités de traitement spécialisées pour accélérer les fonctions intensives en calcul comme l'estimation de mouvement, la DCT.

### 2.1. codeurs dédiés

Les codeurs dédiés offrent une programmabilité réduite, leur architecture est spécialisée pour un standard de codage ou décodage particulier. Il y a eu récemment une prolifération d'annonces de produits en raison de l'arrivée massive d'applications exigeant la compression vidéo. Notons cependant que la plupart des produits sont des décodeurs vidéo, ceux-ci étant plus simples à implémenter car ne nécessitant pas d'estimation de mouvement.

Il convient de remarquer que certains circuits incluent maintenant à la fois des unités programmables et non programmables et peuvent être considérés comme des solutions hybrides de processeurs programmables et spécialisés. Le jeu de circuits de codage d'AT&T [1] pour l'implémentation de systèmes de téléconférence à bas coût et le circuit de codage H.261 de Sgs-Thomson [11] illustrent cette évolution.

### 2.2. processeurs de traitement du signal vidéo

Les VSP ne sont pas dédiés à une application. Leur architecture est organisée autour d'un cœur de processeur pour le calcul classique et peut inclure des unités dédiées pour l'implantation efficace de parties de la chaîne de compression (DCT, estimation de mouvement, etc.). Le problème majeur de telles approches est de supporter l'énorme puissance de calcul requise par les applications. Ce problème est abordé de trois façons différentes.

- Accroissement de la fréquence d'horloge par utilisation intensive du pipeline.
- Exploitation de la distribution de données en implémentant plusieurs chemins de données parallèles sur un même circuit.

- Mise en œuvre des tâches intensives en calcul (estimation de mouvement, DCT) sur des groupes de processeurs (approche coprocesseur).

### 2.2.1. accroissement de la fréquence d'horloge

Les processeurs S-VSP [14] et VSP3 [17] de NEC sont des exemples typiques de processeurs de traitement de signal vidéo. Leur performance résulte de l'utilisation d'une fréquence d'horloge élevée par l'utilisation intensive du pipeline. Le VSP3 est un processeur atteignant les 1,5 GOPS, fonctionnant à 300 MHz, et ayant 1,3 millions de transistors.

### 2.2.2. chemin de données parallèles

Le circuit IDSP [25] des laboratoires NTT est un exemple de VSP mettant en œuvre des chemins de données parallèles. Celui-ci comporte quatre unités de traitement de données pipelinées (910 000 transistors), atteignant ainsi une performance de crête de 300 MOPS à une fréquence de fonctionnement de 25 MHz.

### 2.2.3. approche coprocesseur

Les capacités de traitement du circuit VDSP2 [24] de Matsushita Electric sont dues à l'utilisation d'une architecture SIMD à quatre processeurs DSP contrôlée par un noyau de DSP. Le circuit dispose en plus d'unités spécialisées pour la DCT et le codage à longueur variable. Il fonctionne à 80 MHz et intègre 2,5 millions de transistors.

Le circuit MVP ou TMS320C80 [15] de Texas Instruments est radicalement différent de tous les autres processeurs vidéo. Il n'a aucune unité spécifique. Il inclut un processeur maître (processeur RISC 32 bits à usage général), quatre DSP parallèles, une unité de contrôle de transfert, de la mémoire locale et deux contrôleurs vidéo. Il atteint ainsi une performance de crête de 2 GOPS à une fréquence de fonctionnement de 50 MHz et intègre plus de 4 millions de transistors.

## 2.3. approche Movie

Comme l'illustrent les approches évoquées précédemment, l'emploi du parallélisme et de la spécialisation constitue un moyen efficace pour atteindre la puissance de calcul requise par les algorithmes de traitement vidéo. Les quelques architectures proposées dans la littérature, basées sur un style de parallélisme SIMD [8, 16], ou caractérisées par l'emploi de modèles MIMD [23, 10, 6] illustrent bien ce souci d'associer parallélisme et spécialisation.

Les caractéristiques très différentes de toutes ces approches résultent de compromis tels que le niveau de performance recherché, le coût, le volume, l'extensibilité et l'évolutivité de la solution. Ils relèvent tous de l'adéquation application architecture.

L'aspect d'évolutivité de la solution est fondamental et doit être considéré avec attention. Il est évident que c'est dans la programmabilité que réside la pérennité des choix architecturaux. La programmabilité est la seule possibilité pour offrir un cadre d'expérimentation pour de nouvelles applications, aborder la conception de nouveaux algorithmes sous l'angle de leur mise en œuvre et ainsi contribuer à faciliter leur implantation future dans des systèmes spécialisés [20]. La problématique de la conception s'en trouve néanmoins complexifiée, puisque l'adéquation concerne alors trois relations intimement liées et de même importance : application, architecture, programmation [19].

Le modèle de programmation doit convenir au domaine d'applications et au modèle de machine. Réciproquement, l'architecture parallèle et son modèle d'exécution doivent présenter de bonnes caractéristiques pour supporter efficacement les applications. Il est cependant difficilement concevable que la conception d'une architecture programmable puisse porter sur les aspects algorithmiques du codage vidéo. Il est donc raisonnable de considérer ces aspects comme figés. C'est cette approche qui a guidée la spécification de l'architecture de MOVIE.

Les exigences des algorithmes vidéo-numériques ont été identifiées en examinant différentes techniques de codage par blocs : 34 Mbit/s [12], MPEG-1 [9], MPEG-2 [22], codage en sous-bande. Les différentes tâches algorithmiques (DCT, quantification, estimation du mouvement, codage à longueur variable, etc.) ont été quantifiées en terme de coût et les types de calcul répertoriés. Les propriétés de régularité des algorithmes, tant au niveau des traitements que de l'accès aux données, ont motivé l'utilisation d'un parallélisme de type régulier, SIMD ou systolique, en lien étroit avec les modèles de programmation et d'exécution sous-jacents.

Les études préliminaires ont porté sur l'utilisation de réseaux linéaires SIMD, où chaque colonne de pixels de l'image est allouée à un processeur différent du réseau [7]. Le traitement d'un bloc de  $n \times n$  pixels de l'image est dans ce cas réalisé sur un groupe de  $n$  processeurs du réseau. Certains algorithmes comme la DCT ne réclament que des communications entre processeurs d'un même groupe. D'autres, comme l'estimation de mouvement, exigent en revanche des communications entre groupes. Le besoin de permettre de confiner des traitements au sein d'un groupe a été mis en évidence. Aussi, le modèle d'architecture MOVIE consiste en un réseau linéaire de groupes de processeurs systoliques. Un groupe est constitué d'un petit réseau linéaire de processeurs élémentaires de calcul, dont le modèle d'exécution est systolique SIMD. Ce réseau est associé à un processeur dit d'entrée/sortie, en charge de la gestion de la mémoire de donnée externe au groupe ainsi que de l'alimentation en données du réseau de processeurs de calcul.

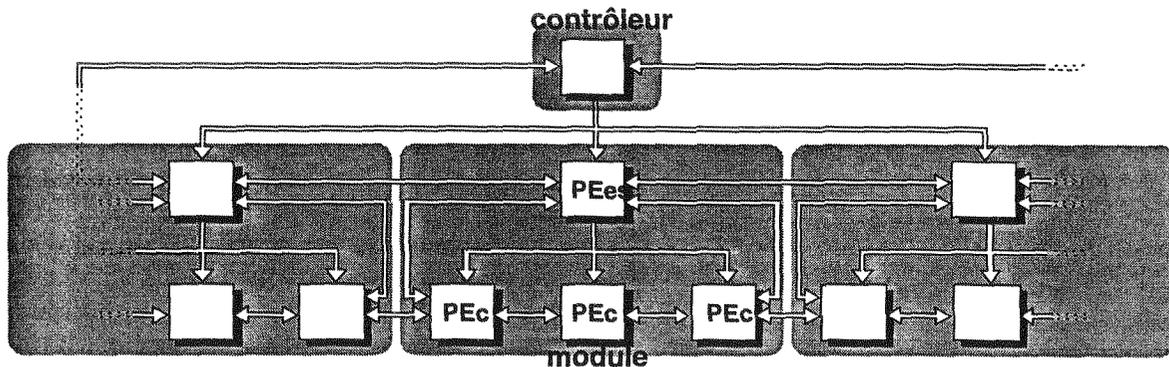


Figure 1. – Différents niveaux de l'architecture MOVIE.

### 3. architecture MOVIE

#### 3.1. différents niveaux de parallélisme dans l'architecture MOVIE

L'association d'un groupe de processeurs de calcul ( $PE_c$ ) et du processeur d'entrée/sortie ( $PE_{es}$ ) constitue la brique de base (ou module) de l'architecture MOVIE. Un module peut être vu comme un petit réseau linéaire de processeurs de calcul relié à ses extrémités à un processeur en charge de l'alimentation en données et/ou de la récupération de résultats. Les modules voisins communiquent par l'intermédiaire de ports de communication à deux niveaux : au niveau des réseaux de processeurs de calcul et au niveau des processeurs d'alimentation en données, comme illustré par la figure 1.

Le réseau de processeurs de calcul est alimenté par le processeur d'entrée/sortie associé, soit par décalage de données, soit par diffusion. Les processeurs d'entrée/sortie sont alimentés par le contrôleur de façon similaire. Ils ont aussi accès à une mémoire de forte capacité, externe au circuit, dans laquelle sont stockées les portions d'images en cours de traitement.

Les traitements tirant parti d'un découpage au niveau du pixel peuvent être effectués par les processeurs de calcul, tandis que ceux plus adaptés à un découpage au niveau du bloc sont réalisés par les processeurs d'entrée/sortie.

#### 3.2. circuit MOVIE

Le circuit MOVIE dispose, du point de vue externe, des ports suivants, comme illustré par la figure 2 :

- quatre ports de données 16 bits pour les échanges entre circuits,
- trois ports vidéo 16 bits pour les entrées/sorties vidéo,
- deux ports d'instructions pour le contrôle du  $PE_{es}$  et des  $PE_c$ ,

- un port d'interface mémoire pour les échanges avec la mémoire externe.

Son architecture interne, illustrée par la figure 2, est organisée autour de quatre éléments principaux qui sont le réseau de processeurs de calcul ( $PE_c$ ), le processeur d'entrée/sortie ( $PE_{es}$ ), les décodeurs d'instructions et le contrôleur vidéo.

Les processeurs  $PE_c$  (16 bits) et  $PE_{es}$  (32 bits), ont sensiblement la même architecture, ce qui a l'avantage de minimiser les développements matériel et logiciel. En dehors de la largeur du chemin de données, le  $PE_{es}$  diffère du  $PE_c$  par sa connexion à une mémoire externe, et par quelques instructions et registres de configuration supplémentaires.

Les opérations du circuit sont contrôlées par deux instructions codées sur 40 bits, engendrées à l'extérieur du circuit et décodées en interne par une unité de décodage. Ce schéma de contrôle tire parti de la régularité de l'architecture où tous les processeurs  $PE_c$  exécutent simultanément la même instruction. Afin d'accroître les performances, l'exécution des instructions est pipelinée : il y a recouvrement entre les phases de décodage et d'exécution,

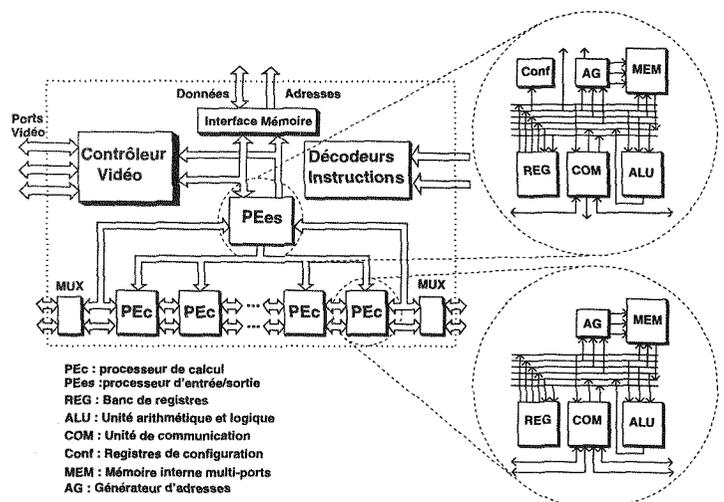


Figure 2. – Architecture générale du circuit MOVIE.

permettant ainsi l'exécution d'une nouvelle instruction à chaque cycle d'horloge.

Le contrôleur vidéo a en charge l'acquisition ou la restitution de données. Le circuit dispose de trois ports utilisables en acquisition ou en restitution et capables de gérer les formats vidéo courants (4:2:0, 4:2:2, 4:4:4, etc.). L'emploi de ces ports n'est pas restreint à la vidéo, ils peuvent être utilisés pour émettre ou recevoir tout autre type de données (jusqu'à 16 bits). Les données acquises ou à restituer par le circuit sont stockées dans la mémoire externe, le contrôleur vidéo n'effectuant qu'un stockage temporaire de ces données à l'intérieur du circuit.

### 3.3. capacités de communication et notion de reconfiguration

Le processeur de calcul est capable d'exécuter une opération de calcul et une opération de communication avec ses voisins en parallèle. Comme dans une architecture VLIW, le mot d'instruction se compose de deux champs indépendants : le premier décrit l'opération arithmétique et le second la communication.

En raison des contraintes technologiques, le circuit MOVIE contient un nombre limité de processeurs PE<sub>c</sub>, égal à un sous multiple de la largeur des blocs de pixels. Par exemple, quatre ou huit PE<sub>c</sub> peuvent convenir. La notion de reconfiguration permet d'agréger des modules pour former un groupe composé d'un PE<sub>es</sub>, et de tous les PE<sub>c</sub> contenus dans les modules. Des modules à quatre PE<sub>c</sub> agrégés par deux forment ainsi des groupes de huit PE<sub>c</sub>. Ce mécanisme matériel permet un réel dimensionnement de l'architecture : adaptation de la taille du réseau de processeurs de calcul à la taille des blocs de pixels.

#### 3.3.1. architecture du processeur de calcul

La partie opérative d'un PE<sub>c</sub> comme illustré par la figure 3 comprend :

- un banc de 32 registres,
- une unité arithmétique et logique (UAL à trois étages de pipeline),
- une mémoire multi-accès de 1K mots de 16 bits (deux accès en lecture, deux accès en écriture) et les générateurs d'adresses associés,
- des registres spécialisés pour l'accès automatique à la mémoire, les communications, la mise au point.

L'architecture des processeurs repose sur le concept de jeux d'instructions RISC avec des extensions pour le traitement vidéo. Elle inclut en particulier des mécanismes d'accumulation et de génération automatique d'adresses, permettant à des opérations courantes en traitement d'image comme  $\sum(a - b)^2$ ,  $\sum |a - b|$  ou  $\sum(a - b) \times c$  d'être lancées à chaque cycle. Une opération de communication entre PE<sub>c</sub> peut avoir lieu en parallèle avec les opérations arithmétiques, pour permettre un équilibre entre les calculs et les communications.

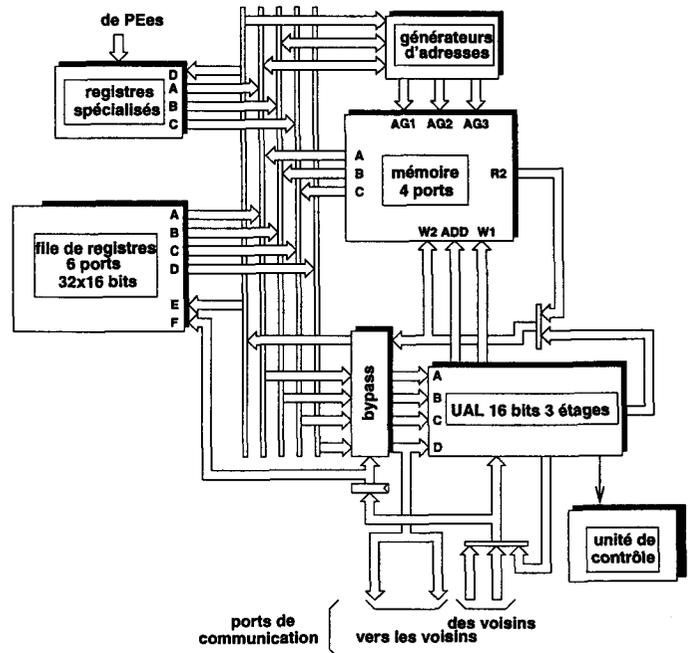


Figure 3. – Architecture interne d'un processeur PE<sub>c</sub>.

## 4. outils de programmation de MOVIE

La programmation d'architectures à base de modules MOVIE a été considérée dès le début de l'étude de l'architecture afin de garantir que les mécanismes architecturaux mis en œuvre étaient exploitables par un compilateur et qu'ils n'étaient pas un obstacle à la production d'un code efficace. La définition de l'environnement de programmation a par conséquent été menée simultanément à la spécification de l'architecture.

L'architecture MOVIE est basée sur un modèle d'exécution SIMD. Dans le modèle de programmation à parallélisme de données qui lui est sous-jacent, le contrôle est séquentiel alors que l'accès aux données est parallèle. Le langage C-stolic [18] dédié à la programmation d'algorithmes réguliers sur une architecture composée d'un réseau linéaire de cellules interconnectées localement et alimentées par un processeur hôte a été employé dans les études préliminaires pour simuler les algorithmes étudiés. Il a maintenant été adapté aux spécificités de l'architecture MOVIE et est employé comme langage de programmation de haut niveau. C-stolic est un langage impératif, à structure de blocs, dont la syntaxe est proche de celle du langage C. Le parallélisme est décrit de façon naturelle et explicite par l'emploi de classes de mémorisation particulières. *stolic* signifie l'allocation sur chaque processeur de calcul d'un exemplaire de la variable,

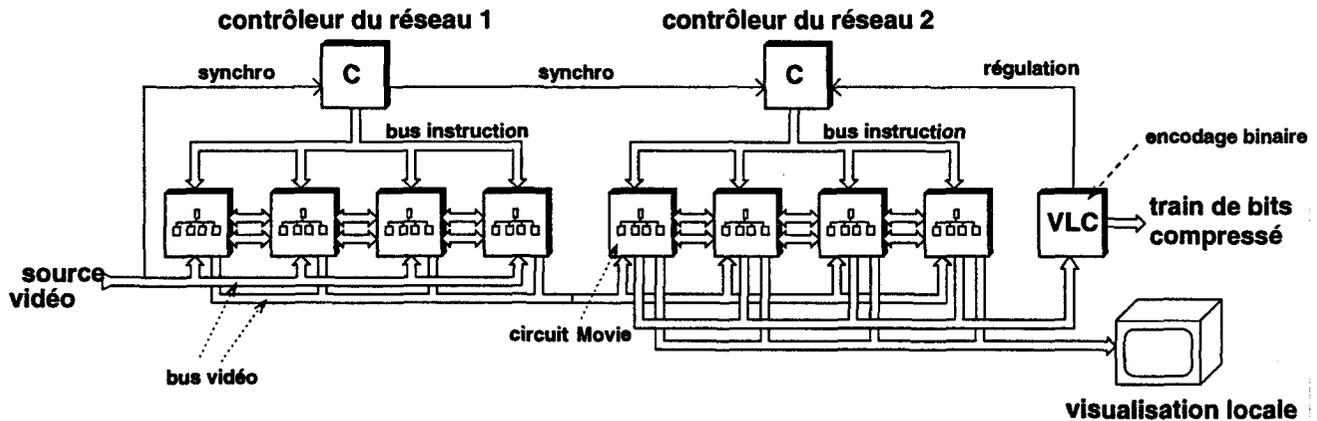


Figure 4. – Codeur MPEG-2 réalisé avec deux réseaux MOVIE.

parallel signifie l'allocation sur chaque processeur d'entrée/sortie d'un exemplaire de la variable, *static* exprime l'allocation de la variable sur le processeur hôte. Un programme écrit en C-stolic est composé d'un seul fichier source duquel sont dérivés le programme des différents processeurs.

Les communications sont synchrones et sont l'effet d'affectations particulières : affectation à droite  $x = > y$ , à gauche  $x = < y$ . Dans ces exemples, chaque processeur émet le contenu de la variable  $y$  et récupère dans  $x$  la donnée provenant de son voisin. Les processeurs du bord ne modifient pas  $x$  ou n'émettent pas  $y$  si le sens de l'affectation ne le permet pas.

Une caractéristique importante des processeurs d'un module MOVIE est de permettre l'exécution en parallèle de plusieurs opérations [2]. La gestion de ce parallélisme interne nécessite la mise en œuvre de techniques d'optimisation de code (optimisation de séquences de code sans branchement, optimisation de boucles par des techniques de pipeline logiciel, de dépliage).

L'approche retenue repose sur l'utilisation d'un outil appelé Oco [5] développé à l'Irisa dans le projet Caps en vue de l'optimisation de code pour des microprocesseurs classiques de type RISC Oco (Optimiseur de Code Objet) inclut des algorithmes d'ordonnancement pour l'optimisation de code sans branchement ainsi que pour les boucles (pipeline logiciel).

La chaîne de compilation gère les différents niveaux de parallélisme d'une architecture à base de circuits MOVIE. Son organisation, consiste en quatre modules principaux s'appuyant sur des outils existants.

- Le compilateur C-stolic traduit le programme source en trois programmes, en langage C ou en format intermédiaire, correspondant aux différents processeurs de l'architecture.
- Le générateur de code traduit les programmes C produits par le compilateur C-stolic en code assembleur séquentiel. La première version du générateur de code, réalisée à l'aide du compilateur multicible *lec* [131], a très largement contribué à la validation de l'architecture interne des processeurs et de leur jeu d'instructions.
- L'optimiseur de code Oco parallélise le code assembleur séquentiel.

- Oco est paramétré par une description de l'architecture cible en terme de types de ressources et d'utilisations des ressources par les instructions. Il est donc possible de modéliser finement l'architecture du processeur et de rendre compte du déroulement des instructions cycle par cycle afin d'optimiser le nombre moyen d'instructions lancées à chaque cycle.

## 5. applications

Un codeur MPEG-2 consiste en trois parties : l'estimation de mouvement, la boucle de codage et la génération du train binaire. L'estimation de mouvement et la boucle de codage manipulent des blocs de pixels de taille  $8 \times 8$ ,  $16 \times 8$  ou  $16 \times 16$ . Ces traitements sont bien adaptés à un parallélisme de données. Les traitements appliqués sur chaque bloc étant très similaires, l'emploi d'un parallélisme de données s'avère être très efficace.

L'implantation la plus efficace d'un codeur MPEG-2 avec des circuits MOVIE consiste à utiliser deux réseaux linéaires interconnectés comme illustré par la figure 4 [3]. Le premier réseau est en charge de l'estimation de mouvement et du réordonnement des trames. Il place sur son flot vidéo de sortie les résultats de l'estimation de mouvement (vecteurs de mouvement). Le second réseau réalise la boucle de codage et émet ses résultats vers l'unité de génération de train binaire. La synchronisation entre les réseaux est réalisée par l'envoi d'un signal du contrôleur du premier réseau vers le second. Cette opération de synchronisation n'est nécessaire qu'en début de chaque trame grâce aux capacités de gestion des contrôleurs vidéo des circuits MOVIE.

Les études d'implantation des algorithmes réalisées montrent que deux réseaux linéaires de 44 circuits MOVIE fonctionnant à 33 MHz sont suffisants pour implémenter un codeur MPEG-2 temps réel complet totalement programmable.

## 6. conception du circuit MOVIE

Le choix du processus de conception du circuit MOVIE a été guidé par plusieurs contraintes ou impératifs, tant techniques qu'humains : permettre une spécification la plus indépendante possible des outils de CAO et de la filière technologique retenue pour la réalisation, explorer différentes alternatives architecturales, et garantir une sûreté de conception.

### 6.1. spécifications

L'analyse de ces contraintes nous a amené à utiliser le langage VHDL pour la spécification du circuit, plutôt qu'une approche plus traditionnelle à base de schématique. VHDL permet de décrire les choix d'implémentation de manière non ambiguë, à un degré d'abstraction suffisant pour pouvoir passer d'une technologie ou d'un outil à un autre avec un minimum de travail de réécriture. Il permet aussi bien d'utiliser une description fonctionnelle que structurelle, nécessaire si l'on veut pouvoir utiliser efficacement les spécificités d'une bibliothèque.

L'utilisation de VHDL et de la synthèse, pour un circuit de la complexité de MOVIE, permet au concepteur de s'affranchir des détails de conception et de travailler sur une description de taille réduite et humainement maîtrisable. L'automatisation du processus de production, à partir de la description VHDL, permet au concepteur de concentrer ses efforts sur les études architecturales.

### 6.2. structuration

La forte structuration du langage VHDL est tout à fait adaptée aux contraintes imposées par le cahier des charges. Une approche de conception descendante nous a permis de structurer de manière hiérarchique le circuit en un certain nombre d'unités fonctionnelles, ou entités VHDL, dont certaines sont utilisées en plusieurs exemplaires. Une fois la vision externe de ces entités définies ainsi que leurs interactions, le concepteur peut se consacrer à leur implémentation, en explorant différentes réalisations possibles sur des critères de performance et de surface. Les éléments de la bibliothèque MOVIE ainsi développés comportent différentes vues : VHDL développées sous SYNOPSIS et physiques développées sous COMPASS.

### 6.3. méthodologie

L'organisation de l'architecture interne des processeurs de MOVIE met en évidence le besoin des unités fonctionnelles suivantes : UAL à trois niveaux de pipeline, mémoire à quadruple

accès, banc de registres à quatre ports en lecture et deux en écriture, générateurs automatiques d'adresses, contrôleur vidéo.

La réalisation de telles unités à partir d'une bibliothèque existante peut se faire selon plusieurs techniques : synthèse logique, synthèse de chemin de données, compilateurs de RAM, de multiplexeurs, instantiation de cellules de la bibliothèque, synthèse spécifique pour les structures de mémorisation. Pour les mémoires et les registres à accès multiples, des techniques de multiplexage permettant deux accès par cycle d'horloge ont été utilisées pour pallier les insuffisances de la bibliothèque.

Par ailleurs à ce stade des spécifications l'impact du partitionnement en un ou plusieurs chemins de données a été évalué. De même, la manière d'écrire le code VHDL de la spécification comportementale intervient sur le processus de synthèse et donc sur l'architecture produite. Pour un composant donné, plusieurs descriptions synthétisables, mettant en oeuvre ces différentes techniques ont été développées et comparées en terme de performance et de surface.

Pour cela les outils classiques d'un logiciel de CAO de circuits ont été utilisés : synthétiseur, optimiseur, vérificateur de « timing », placement/routage pour l'évaluation en surface.

Après plusieurs itérations sur la chaîne spécification-synthèse-évaluation, une bibliothèque de composants, VHDL et physiques est disponible. Les composants sont utilisés dans une description VHDL hiérarchique synthétisable, modélisant les  $PE_c$ ,  $PE_{es}$ , et le module MOVIE. Ces composants de plus haut niveaux sont évalués et améliorés selon les mêmes techniques.

### 6.4. test

L'éditeur graphique de SYNOPSIS est utilisé pour produire la hiérarchie VHDL du niveau le plus haut (module MOVIE), au niveau le plus bas (composants), à partir d'une représentation graphique utilisable lors de la phase de test et de mise au point. Aux entités de plus bas niveau sont associées des architectures qui correspondent à la spécification VHDL synthétisable retenue pour les composants.

Une première étape dans le test, consiste à tester chaque composant en l'incluant dans un banc de test spécifique, écrit en VHDL, générant des stimuli de manière à mettre en oeuvre tous les mécanismes propres aux composants et simulant son environnement. L'analyse des résultats se fait avec le metteur au point. On arrive ainsi à éliminer la majeure partie des erreurs de conception des composants.

Le test des jeux d'instructions et de par la même le test des  $PE_c$ ,  $PE_{es}$  et du module MOVIE se fait avec un environnement de test spécifique. A chaque cycle, trois instructions concernant respectivement les  $PE_c$ , le  $PE_{es}$  et le contrôleur sont lues dans un fichier assembleur produit soit manuellement soit à l'aide du compilateur. Il est ainsi possible par un choix et un ordonnancement judicieux des instructions, de tester chaque composant ainsi que

les détections de conflit d'accès à des ressources, l'exécution conditionnelle et les mécanismes de mise au point. Le résultat de ces jeux de test peut être sauvegardé pour servir ultérieurement de référence à de nouveaux tests en cas de modification des spécifications.

Cette méthodologie de test est aussi bien valable pour les tests fonctionnels, portant sur la description synthétisable, que pour les tests structurels réalisés avec les listes d'interconnexion VHDL rétroannotées avec les paramètres électriques réels après synthèse et routage. Seul change l'architecture associée à l'entité VHDL du composant.

### 6.5. réalisation du circuit

Le circuit MOVIE a été réalisé dans une technologie Cmos  $0.6\mu$  à deux niveaux de métallisation. Dans cette technologie, le processeur  $PE_c$  a une surface de  $20\text{ mm}^2$ , et compte 214 000 transistors, tandis que le  $PE_{es}$  a une surface de près de  $46\text{ mm}^2$  pour 452 292 transistors.

Un circuit MOVIE à quatre processeurs  $PE_c$  occupe une surface de  $367\text{ mm}^2$ , et contient 1,9 millions de transistors (le cœur représente  $302\text{ mm}^2$ ). Le nombre de broches est de 334 (incluant 296 broches de signaux). La consommation estimée est de 4 W à 20 MHz et 5 V.

## 7. conclusion

Les recherches présentées dans ce papier, menées dans le cadre d'une collaboration entre le CCETT et l'Irisa, montrent que l'architecture MOVIE constitue une solution particulièrement flexible pour la réalisation d'un environnement temps réel de traitement vidéo. Les choix architecturaux qui ont guidé la phase de spécification de cette architecture résultent d'une part de l'étude des différents algorithmes qui composent les schémas de compression standardisés actuels et d'autre part de notre expertise en matière de conception d'architectures parallèles spécialisées, en particulier des architectures systoliques.

Les mécanismes architecturaux mis en œuvre dans le module MOVIE permettent la réalisation d'architectures parallèles spécialisées et programmables de configuration différente, bien adaptées à l'implantation de schémas de compression par blocs de type MPEG-1, MPEG-2. L'étude décrite dans [20] portant sur la définition d'une architecture pour des algorithmes d'estimation de mouvement bloc-récurrents montre également l'intérêt du style d'architecture de MOVIE pour des algorithmes moins réguliers quant aux accès aux données.

• Les modules MOVIE peuvent être agencés en réseau linéaire, ou en grille 2D, par simple assemblage des circuits. Plusieurs réseaux, communiquant par l'intermédiaire des ports d'entrée/

sortie vidéo, peuvent par exemple être constitués pour une mise en œuvre pipeline de l'application [2].

• Les architectures ainsi constituées sont principalement réalisées à partir des circuits MOVIE et de leurs mémoires externes. L'intégration dans les circuits des mécanismes d'entrée/sortie vidéo, de possibilités de communications entre processeurs, contribue à la simplicité et à la compacité des solutions.

• Grâce aux possibilités de configurabilité des ports d'entrée/sortie du circuit MOVIE, des groupes de modules peuvent être aisément constitués à partir de plusieurs circuits. Ces mécanismes ont pour but, d'une part de permettre un réel dimensionnement des architectures, et d'autre part de préserver les mécanismes de communication supportés par le circuit.

### BIBLIOGRAPHIE

- [1] Ackland (A.). - The role of VLSI in multimedia. *IEEE Journal of Solid-State Circuits*, vol. 29, n° 4, avril 1994, pp. 381-388.
- [2] Barzic (R.), Bouville (C.), Charot (F.), Le Fol (G.), Lemonnier (P.) et Wagner (C.). - Movie: A building block for the design of realtime simulator of moving pictures compression algorithms. In: *International Conference on Application-Specific Array Processors - ASAP'9* pp. 193-202. - Strasbourg, France, juillet 1995.
- [3] Barzic (R.), Bouville (C.), Charot (F.), Le Fol (G.), Lemonnier (P.) et Wagner (C.). - MOVIE : A Hardware Building Block for Software-Only Real Time Video Processing. In: *IS&T/SPIE's Symposium on Electronic Imaging : Science & Technology, Video Compression : Algorithms and Technologies 1996*. - San Jose, février 1996.
- [4] Baskaran (V.) et Konstantinides (K.). - *Image and Video Compression Standards : Algorithms and Architecture*. - Kluwer Academic Publishers, 1995.
- [5] Bodin (F.), Le Fol (G.) et Raimbault (F.). - *OCO : manuel de l'utilisateur (version préliminaire)*. - Publication interne n° 930, Irisa, mai 1995.
- [6] Bouville (C.), Bouriel (A.), Brusq (R.) et Houlier (P.). - A real-time testbed for MPEG-1 video compression. In: *Visual Communication and Image Processing, SPIE*, pp. 205-212. - Cambridge, Usa, novembre 1993.
- [7] Charot (F.) et Wagner (C.). - Analyse architecturale d'applications de compression de séquences d'image. In : *Quatorzième colloque GRETSI*, pp. 1059-1062.- Juan-Les-Pins, septembre 1993.
- [8] Chin (D.), Passe (J.), Bernard (F.), Taylor (H.) et Knight (S.). - The Princeton engine : a real-time video system simulator. *IEEE Transactions on Consumer Electronics*, vol. 34, n 2, mai 1988, pp. 285-297.
- [9] D. Le Gall. - MPEG: a video standard for multimedia applications. *Communications of the ACM*, vol. 34, n°4, avril 1991, pp. 46-58.
- [10] de Sa (L.), Silva (V.), Perdigao (F.), Faria (S.) et Assuncao (P.). - Parallel architecture for real-time video communication. In : *Vol. 1 60 Visual Communications and Image Processing, SPIE*, pp. 38-387.- novembre 1990.
- [11] Harrand (M.) et al. - A single chip videophone video encoder/decoder. In : *Proceedings IEEE International Solid-State Circuit conference* pp. 292-293. - février 1995.
- [12] ETSI. - Specification of component TV codec for 32-45 Mbit/s. - ETSI Technical report, décembre 1990.

- [13] Fraser (C.) et Hanson (D.). - *A retargetable C compiler: design and implementation*. - The Benjamin/Cummings Publishing Company, Inc., 1995.
- [14] Goto (J.), Ando (K.), Inoue (T.), Yamashina (M.), Yamada (H.) et Enomoto (T.). - 250-MHz BiC- MOS Super-High-Speed Video Signal Processor (S-VSP) ULSI. *IEEE Journal of Solid-State Circuits*, vol. 26, n°12, dec 1991, pp. 1876-1884.
- [15] Gutttag (K.), Gove (R. J.) et Van Aken (J.R.). - A Single-Chip Multiprocessor For Multimedia : The MVP. *IEEE Computers Graphics and Applications*, novembre 1992, pp. 53-64.
- [16] Hoek (C.), Heiss (R.) et Mueller (D.). - An array processor approach for low bit rate video coding. *Signal Processing: Image Communication 1*, vol. 1, n°2, octobre 1989, pp. 213-283.
- [17] Inoue (T.), Goto (J.), Yamashina (M.), Suzuki (K.), Nomura (M.), Koseki (Y.), Kimura (T.), Motomura (M.), Shih (B.S.), Horiuchi (T.), Hamatake (N.), Ku- magai (K.), Enomoto (T.) et Yamada (H.). - A 300- MHz 16-b BiCMOS Video Signal Processor. *IEEE Journal of Solid-State Circuits*, vol. 28, n° 12, décembre 1993, pp. 1321-1329.
- [18] Lavenier (D.), Quinton (P.) et Raimbault (F.). - Architectures systoliques et parallélisme de données : L'environnement de programmation RELACS. *Technique et science informatiques*, vol. 12, n 5, décembre 1993, pp. 597-620.
- [19] Le Fol (G.). - *Architecture parallèle pour le traitement vidéo : contribution à la conception d'un module VLSI programmable et à l'étude d'outils de compilation reciblables*. - thèse, université de Rennes 1, septembre 97.
- [20] Lemonnier (P.). - *Étude d'une architecture parallèle VLSI pour un algorithme d'estimation de mouvement bloc-récurusif*. - thèse, université de Rennes 1, juin 1996.
- [21] Pirsch (P.), Demassieux (N.) et Gehrke (W.). - VLSI Architectures for Video Compression-A Survey. *Proceedings of the IEEE*, vol. 83, n°2, février 1995, pp. 220-246.
- [22] Puri (A.). - Video coding using the mpeg-2 compression standard. *In : SPIE*, pp. 1701-1713. - 1993.
- [23] Tamitani (I.), Harasaki (H.), Nishitani (T.), Endo (Y.), Yamashina (M.) et Enomoto (T.). - A real-time video signal processor suitable for motion picture coding applications. *IEEE Transactions on Circuits and Systems*, vol. 36, n°10, octobre 1989, pp. 1259-1266.
- [24] Toyokura (M.), Kodama (H.), Miyagoshi (E.), Okamoto (K.), Gion (M.), Minemaru (T.), Ohtani (A.), Araki (T.), Takeno (H.), Akiyama (T.), Wilson (B.) et Aono (K.). - A Video DSP with a Macroblock-Level Pipeline and a SIMD Type Vector Pipeline Architecture for MPEG2 CODEC. *IEEE Journal of Solid-State Circuits*, vol. 29, n° 12, décembre 1994, pp. 1474-1481.
- [25] Yamauchi (H.), Tashiro (Y.), Minami (T.) et Suzuki (Y.). - Architecture and Implementation of a Highly Parallel Single-Chip Video DSP. *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 2, n° 2, jun 1992, pp. 207-220.

**Manuscrit reçu le 23 juillet 1997.**

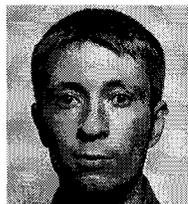
## LES AUTEURS

### François CHAROT



François Charot est docteur de l'université de Rennes 1 (1984). Depuis 1986, il est chargé de recherche Inria à l'Irisa dans le projet *Architectures Parallèles Intégrées*. Ses centres d'intérêt concernent la conception et la réalisation d'architectures parallèles spécialisées, pour des applications liées à l'image, et notamment dans le domaine du traitement numérique du signal vidéo, la compilation pour les processeurs spécialisés programmables (ASIP) et les outils pour la conception de systèmes matériels.

### Gwendal LE FOL



Gwendal Le Fol a obtenu un doctorat de l'Université de Rennes 1 en 1997. Sa thèse s'est déroulée au sein du projet *Architectures Parallèles Intégrées* de l'Irisa, elle avait pour objet la définition de la chaîne de compilation de MOVIE dans une perspective de co-conception matériel-logiciel. Gwendal Le Fol est actuellement ATER à l'Ifsic (Université de Rennes I).

### Charles WAGNER



Charles Wagner a obtenu son diplôme d'ingénieur ENSERG en 1973. Actuellement il est ingénieur de recherche CNRS à l'Irisa. Ses centres d'intérêt principaux sont le codesign, les circuits asynchrones ainsi que les outils de CAO.