

Rapport INRIA 1994 — Programme 1
Architectures parallèles intégrées

Projet API

3 mai 1995

Projet API

Architectures parallèles intégrées

Localisation : *Rennes*

Mots-clés : API (1), architecture parallèle (1, 13), architecture systolique (1, 6, 12, 13), ASIC (1, 9, 11), biologie moléculaire (12), CAO (1, 5, 8, 9, 11), code correcteur (14), compression (14), conception conjointe (5), corps finis (14), cryptage (14), environnement de programmation (6), génome (1, 12), langage fonctionnel (8), langage parallèle (6), langage synchrone (5), optimisation de code (7), simulation d'architecture (1, 5), synthèse d'architecture (1, 8, 9), traitement du signal vidéo (1, 13), VLSI (1, 5, 11, 13, 14).

Api est un projet commun Inria/CNRS (URA 227).

1 Composition de l'équipe

Responsable scientifique

Patrice Quinton, DR CNRS

Secrétaire

Maryse Auffray, AA Inria

Personnel Inria

François Charot, CR

Éric Gautrin, IR (Atelier)

Hervé Le Verge, CR

Tanguy Risset, CR à partir du 1^{er} octobre 1994

Personnel Ura 227

Rumen Andonov, Ater Ifsic jusqu'au 30 août 1994
Dominique Lavenier, CR CNRS
Laurent Perraudou, Ater, maître de conférences Ifsic à partir
du 1^{er} septembre 1994
Sanjay Rajopadhye, professeur associé Insa, CR CNRS à partir
du 1^{er} octobre 1994
Yannick Saouter, CR CNRS
Charles Wagner, IR CNRS (Atelier)

Chercheurs invités

Doran Wilde, université d'Oregon jusqu'au 30 juin 1994

Chercheurs doctorants

Ronan Barzic, bourse CCETT
Florent Dupont de Dinechin, AMN
Pascale Guerdoux, bourse MESR à partir du 1^{er} novembre
1994
Alain Kérihuel, bourse MESR jusqu'au 31 août 1994, Ater Ifsic
ensuite
Gwendal Le Fol, bourse MESR
Pascal Lemonnier, bourse MESR
Patricia Lemoënner, bourse Inria à partir du 1^{er} octobre 1994
Roderick McConnell, bourse Cies jusqu'au 30 septembre 1994
Sophie Robert, bourse Inria à partir du 1^{er} novembre 1994
Oumarou Sié, bourse Cies
Marc Vieillot, bourse MESR

Collaborateurs extérieurs

Catherine Dezan, Maître de conférences à l'UBO
Frédéric Rimbault, Ater à l'IUP de Vannes

In memoriam

Hervé Le Verge est décédé accidentellement le 27 février 1994. Après avoir fait sa thèse dans le projet Api, il avait été recruté comme chercheur Inria en octobre 1993. Les recherches menées par Hervé Le Verge ont porté essentiellement sur la synthèse d'architecture systolique à partir du langage Alpha. La première version du logiciel Alpha du Centaur

lui est due. Il a aussi réalisé la première mise en œuvre de la librairie de calcul sur les polyèdres convexes Polylib, mise dans le domaine public, et utilisée par de nombreuses équipes de recherches, pour la synthèse d'architecture, la vérification de programmes, ou la résolution de problèmes combinatoires. Les travaux de recherches d'Hervé Le Verge ont donné lieu à plusieurs publications de très grande valeur. Grâce à ses qualités scientifiques et humaines, il avait noué de nombreuses relations dans la communauté nationale et internationale.

2 Présentation générale et objectifs

La densité des circuits intégrés permet de réaliser des systèmes de traitement de l'information d'une très grande complexité dans un volume de matériel faible. Dans beaucoup de domaines d'application, il est impératif de mettre en œuvre certains algorithmes, voire des systèmes entiers, directement sur silicium. Pour aboutir, une telle démarche requiert une grande maîtrise des différentes étapes de la conception des systèmes matériels, depuis la spécification de l'application envisagée jusqu'à la production des composants, leur assemblage et leur test.

La réalisation d'architectures spécifiques nécessite l'emploi de techniques variées – spécification, simulation, vérification, validation, optimisation – dans le but de produire rapidement une implantation matérielle correcte. Les recherches abordées dans le projet Api concernent les méthodes de conception et de réalisation d'architectures intégrées pour des applications spécifiques, pour des domaines d'application relevant essentiellement du traitement du signal et de l'image. Il s'agit d'être capable de produire des systèmes matériels exempts de défauts, satisfaisant aux performances requises pour l'application, dans un délai minimum. En outre, les méthodes étudiées doivent être indépendantes des technologies physiques de réalisation, et doivent pouvoir s'intégrer à des environnements et des langages de conception largement répandus. Le but du projet est d'assembler de façon cohérente des outils pour la spécification, la simulation – si possible temps réel – et la réalisation architecturale d'une application, permettant ainsi de valider des choix algorithmiques avant la mise en œuvre matérielle proprement dite [6].

Plusieurs remarques guident notre recherche.

- Pour parvenir à traiter correctement la spécification et la synthèse d'architectures parallèles, il faut s'appuyer sur des modèles formels

solides. Ceci justifie les recherches menées dans Api sur le langage Alpha fondé sur les équations récurrentes.

- Une application de traitement de signal ou d'image un tant soit peu complexe se décompose en deux niveaux : des blocs fonctionnels intensifs en calcul, et un niveau de contrôle entre ces blocs, peu exigeant en volume de calculs, mais très complexe en synchronisation du fait des nombreuses interactions entre blocs. Cette remarque conduit à utiliser des formalismes différents pour chaque niveau, et nous conduit à étudier l'utilisation conjointe de langages flot de données synchrones (Signal par exemple) pour le contrôle et Alpha pour les calculs intensifs.
- La conception d'une architecture parallèle comporte de nombreuses étapes, et exige des simulations très coûteuses en temps, le plus souvent irréalisables sur des machines d'usage général. Ceci nous a conduit à imaginer des architectures semi-spécialisées réalisées à partir de briques de base matérielles et logicielles et permettant la mise en œuvre rapide d'applications, pour les besoins de la simulation. L'étude du langage C-stolic et de sa compilation sur des architectures diverses se place dans ce contexte. La conception de processeurs pour la compression d'image, ou pour les calculs du génome vont aussi dans ce sens.
- Enfin, il ne saurait y avoir de méthodologie sans une expérience sérieuse d'applications réelles, en lien avec les concepteurs d'architectures. C'est pour cette raison qu'une part importante de l'activité du projet Api est consacrée à l'étude d'applications, essentiellement dans le domaine du traitement d'image et de la biologie moléculaire.

3 Actions de recherche

Les recherches s'articulent autour des trois actions suivantes :

- l'étude d'outils pour la conception et la simulation d'architectures spécialisées,
- l'étude des techniques de synthèse d'architectures régulières,
- l'étude d'applications.

3.1 Outils pour la conception et la simulation d'architectures spécialisées

Depuis 1991, le langage Signal développé à l'Irisa dans le projet EP-ATR est utilisé par le projet Api pour la modélisation de systèmes de traitement de signaux. Signal est intéressant pour exprimer le niveau contrôle de l'application. À terme, l'objectif est de s'appuyer sur ce langage, ou plus généralement, sur des langages flots de données synchrones, pour spécifier les applications en vue de leur mise en œuvre sur des architectures de différents types : des machines parallèles à usage général afin d'effectuer une validation des spécifications, des machines spécialisées configurables destinées à effectuer une validation fine de l'application sous contraintes de temps réel et, enfin, des machines spécialisées satisfaisant aux contraintes de performance, d'encombrement, de coût et faisant intervenir des circuits VLSI.

Les algorithmes réguliers, intensifs en calcul et hautement parallélisables, peuvent être programmés dans le langage à parallélisme de données C-stolic, et leur mise au point réalisée à l'aide de l'environnement logiciel Relacs. Dans le futur, il sera possible d'utiliser aussi le langage Alpha décrit dans le paragraphe 3.2, mais l'environnement de synthèse basé sur Alpha n'a pas encore actuellement atteint un degré de maturité suffisant pour être utilisé à cet effet.

Les recherches menées sur ce sujet en 1994 portent sur les thèmes suivants :

- l'étude et le développement d'un environnement de prototypage de systèmes matériels à l'aide de langages synchrones ;
- l'étude de l'environnement de programmation de machines de type SIMD Relacs ;
- l'étude de la génération de code optimisé pour des processeurs spécialisés programmables ;
- l'étude de mise en œuvre de Gamma sur une structure matérielle à base de circuits logiques programmables.

3.1.1 Prototypage de systèmes matériels à l'aide de langages flots de données synchrones

Participants : François Charot, Alain Kérihuel, Roderick McConnell, Patrice Quinton, Sanjay Rajopadhye

Les langages flots de données synchrones sont de plus en plus souvent utilisés pour spécifier et mettre en œuvre des systèmes spécialisés logiciels. Le but de cette recherche est d'adapter ce type de méthode à la conception de systèmes matériels, et en particulier, de prendre en compte la description de composants déjà existants.

Le modèle SDF (flots de données synchrones) de Lee et Messerschmitt a été pris pour point de départ, et adapté au cas d'horloges *périodiques*. Un calcul sur ce type d'horloges a été défini. Il permet d'assembler des composants ayant des caractéristiques différentes, et de déterminer leur synchronisation. Ce modèle est la base d'une méthode de conception qui peut être utilisée en Signal et en VHDL pour concevoir des systèmes matériels. Cette méthode a été appliquée à l'étude d'une chaîne de codage vidéo (thèse de R. McConnell, [1]).

Par ailleurs, la relation entre les langages Alpha et Signal a été étudiée. L'objectif est de pouvoir, à terme, spécifier une application complète avec un mélange cohérent de ces deux langages. L'interface doit se faire d'une part, en étendant la dimension temporelle des expressions Alpha à des domaines périodiques, et d'autre part, en détectant des ensembles de variables Signal dont les horloges sont aussi de type périodiques. Cette étude servira de base au projet Cairn avec France-Télécom.

Enfin, l'étude du portage de SynDEx, développé à l'Inria Rocquencourt dans le projet Sosso (Y. Sorel), sur la machine iWarp a été réalisée. L'environnement SynDEx réalise l'implantation d'algorithmes exprimés sous forme de graphes de flots de données conditionnés sur des architectures parallèles, en générant le code destiné à la machine cible. Le graphe peut être issu d'une spécification écrite à l'aide du langage Signal.

3.1.2 Environnement de programmation pour algorithmes systoliques

Participants : Dominique Lavenier, Frédéric Raimbault

Dans un système de traitement spécialisé, les algorithmes intensifs en calcul doivent être parallélisés. Se pose alors le problème de leur simulation pour les valider et les évaluer, mais aussi pour s'assurer que le comportement fonctionnel de l'application complète est correct. Une alternative est la synthèse automatique, abordée dans le paragraphe 3.2, mais cette technique n'est pas encore assez mûre pour remplacer l'étude directe des algorithmes parallèles. Le dernier type de validation nécessite souvent la simulation en temps réel – par exemple, la compression d'images où la qualité est un élément essentiellement subjectif.

Relacs est un environnement de programmation destiné à répondre à ces besoins : son but est de permettre la programmation d'algorithmes parallèles de type systolique et leur exécution sur des machines – séquentielles ou parallèles – disponibles commercialement (thèse de F. Raimbault, [2]). Le langage de programmation de Relacs est appelé C-stolic. C'est un langage à parallélisme de données proche du langage C, qui permet de décrire les flots de données caractéristiques des algorithmes systoliques et de résoudre simplement les problèmes d'entrées/sorties, cruciaux dans ce type d'architectures. Le compilateur génère un code distinct pour le processeur d'entrées/sorties, pour les processeurs du réseau et pour la machine hôte à partir de la répartition explicite fournie par le programmeur. Toutes les difficultés liées à la programmation par processus communicants sont résolues par le schéma de compilation.

Relacs permet l'exécution d'algorithmes systoliques sur de nombreuses architectures parallèles. Le compilateur génère des programmes C fonctionnellement équivalents aux processus d'entrées/sorties (processeur d'E/S) et de calcul (processeurs du réseau). Ceux-ci peuvent être compilés sur différentes machines parallèles telles que l'IPSC2, un réseau de cellules iWarp ou la machine ArMen de l'UBO.

L'environnement Relacs possède un compilateur pour station de travail permettant l'étude et la mise au point de nouveaux algorithmes. Un débogueur graphique aide l'utilisateur dans sa démarche. La génération de code pour des architectures spécialisées, notamment à base de logique reconfigurable, est en cours d'étude.

3.1.3 Génération de code optimisé pour des processeurs spécialisés programmables

Participants : François Charot, Gwendal Le Fol

Les études de conception d'un processeur spécialisé programmable, brique de base pour la réalisation de simulateurs temps-réel d'applications de compression de séquences d'images ont été accompagnées de recherches sur la définition d'une chaîne de compilation, allant du programme C-stolic à un code optimisé, paramétrable en fonction des caractéristiques fines de l'architecture du processeur.

Au cours de l'année 1994, nous nous sommes focalisés sur le problème de l'optimisation de code au niveau bas. Nous avons expérimenté le logiciel Oco développé à l'Irisa dans le projet Caps. À partir d'un code sous forme assembleur décoré (c'est à dire incluant les informations de dépendance), Oco génère un code assembleur optimisé. L'ensemble des étapes de transformation et en particulier la phase d'ordonnancement des instructions est entièrement paramétrée par une description de l'architecture cible. Le processeur iWarp, d'Intel, a servi de support à cette expérimentation. La modélisation de son jeu d'instructions – celui-ci comporte en particulier une instruction “parallèle” exploitant les multiples unités fonctionnelles du processeur – a permis de mieux cerner l'intérêt de l'outil Oco ainsi que les extensions et améliorations qui doivent y être apportées. Ces études font l'objet d'une étroite collaboration avec François Bodin du projet Caps.

Cette étude est intéressante et importante à plusieurs titres. Elle nous permet d'une part, d'avoir une bonne compréhension des algorithmes d'optimisation mis en œuvre dans Oco, ainsi qu'une bonne maîtrise de la modélisation de l'architecture d'un processeur. Elle est d'autre part riche d'enseignement quant à la façon d'intégrer un tel outil dans une chaîne complète de compilation pour des processeurs spécialisés.

3.1.4 Mise en œuvre de Gamma sur Perle

Participants : Dominique Lavenier, Marc Vieillot

Le modèle Gamma, développé à l'Irisa par le projet Lande, repose sur une structure de données unique, le multi-ensemble, et les calculs s'apparentent à une suite de réactions *chimiques* consommant ou produisant

des éléments en fonction de règles spécifiques. Le parallélisme intrinsèque de Gamma ne peut être exploité pleinement sur une machine conventionnelle, d'où l'idée de recourir à une architecture de machine adaptée. C'est cette idée qui a conduit à étudier la mise en œuvre sur le processeur PAM (*Programmable Active Memory*) développé par J. Vuillemin du centre de recherche DEC-PRL.

Le processeur PAM est constitué principalement de circuits logiques programmables (FPGA : *Field Programmable Gate Array*). Il permet ainsi de câbler par programmation des structures logiques simples (additionneurs, registres, ...) et de les interconnecter pour former ensuite des architectures logiques plus complexes. Sur un même support physique différentes architectures de machines peuvent ainsi être développées et testées.

L'étude actuelle à laquelle participe Jean-Pierre Banâtre a pour but d'étudier les possibilités de mise en œuvre de Gamma sur une telle structure [13]. Plus généralement, les problèmes de synthèse d'architectures sur FPGA, de stratégie et de méthodologie de conception sont étudiés.

3.2 Synthèse d'architectures parallèles régulières

Dans la plupart des systèmes de traitement spécialisés, les algorithmes intensifs en calcul, c'est-à-dire ceux qui sont candidats à une mise en œuvre parallèle et intégrée, sont de nature très régulière. L'étude du passage d'un algorithme régulier à sa version parallèle est un domaine de recherche très actif au niveau international, dans lequel notre projet a une place reconnue. Sur le plan technique, il s'agit de spécifier l'algorithme considéré à l'aide d'un *système d'équations récurrentes*, puis, par raffinements progressifs, à amener ce système sous une forme qui correspond directement à son exécution sur une architecture régulière. À partir du résultat, on peut alors soit produire un circuit, soit engendrer du code pour une machine parallèle.

3.2.1 Alpha et la synthèse d'architectures régulières

Participants: Rumen Andonov, Catherine Dezan, Florent de Dinechin, Hervé Le Verge, Patrice Quinton, Frédéric Raimbault, Sanjay Rajopadhye, Tanguy Risset, Yannick Saouter, Doran Wilde

Notre effort est concentré sur le développement d'un logiciel pour la synthèse d'architectures régulières. Il permet la dérivation d'algorithmes réguliers – notamment systoliques – par transformations interactives. Le langage Alpha [44] est la base de ce logiciel. Il permet à la fois l'expression d'un algorithme parallèle régulier, et la description d'une architecture synchrone qui en permet l'exécution.

Nos recherches en 1994 ont porté sur les aspects suivants :

- l'extension du langage Alpha pour permettre en particulier la modularité ;
- la conception de nouveaux algorithmes pour les transformations de parallélisation ;
- la mise en œuvre de l'environnement Alpha sous Mathematica, appelé MMAAlpha ;
- enfin, de nombreux exemples d'algorithmes réguliers ont été étudiés.

La définition du langage Alpha inclut la notion de sous-système, mais cette notion ne permet pas de représenter la modularité telle qu'elle est souhaitable au cours de la synthèse. Cette situation a conduit à introduire un nouvel opérateur dans le langage Alpha, appelé *opérateur de décomposition*, qui permet de considérer de diverses façons le type d'une expression multidimensionnelle. Ainsi, une matrice peut-être décomposée sous forme d'un vecteur de vecteurs. Avec cet opérateur, il est possible de définir de façon très puissante des extensions polymorphes des sous-systèmes Alpha. La sémantique de cette extension de Alpha est en cours de définition, ainsi que l'extension des règles axiomatiques qui permettent les transformations de synthèse.

Les transformations de parallélisation font appel à des algorithmes de calcul sur les polyèdres souvent difficiles à mettre au point. Ces algorithmes sont du reste largement utilisés dans les paralléliseurs de programmes impératifs, d'où leur intérêt. Au cours de 1994, deux problèmes ont été étudiés [43]. D'une part, le calcul de la projection d'un

polyèdre qui minimise le nombre de points du projeté, par des méthodes heuristiques. Cet algorithme est important pour déterminer de bonnes fonctions de placement des algorithmes. D'autre part, la génération de boucles énumérant les points d'un polyèdre, algorithme fondamental pour produire le programme ou l'architecture résultant de la parallélisation.

Enfin, le développement de MMAAlpha s'est poursuivi. Outre la consolidation des transformations déjà réalisées, plusieurs nouveaux logiciels ont été mis au point. Un traducteur de Alpha vers C permet désormais d'"exécuter" des systèmes d'équations, suivant une stratégie "à la demande" classiquement employée pour les langages fonctionnels. Un mécanisme de mémorisation évite les calculs inutiles. Un simulateur graphique symbolique peut être utilisé pour simuler des réseaux réguliers. Enfin, un traducteur de Alpha en VLSI a été mis au point, pour les programmes pouvant être interprétés comme des architectures. Le résultat de cette traduction est interprété par le logiciel Madmacs pour générer un circuit régulier (voir paragraphe 3.2.2).

Les recherches sur les algorithmes réguliers sont extrêmement profitables pour l'élaboration de modèles et de techniques de synthèse. L'algorithme du "sac-à-dos" sert depuis deux ans de base pour l'étude des récurrences dépendantes des données. Au cours de 1994, deux nouvelles versions d'architectures linéaires pour ce problème ont été imaginées [28]. La première version est un algorithme systolique pur, le seul à notre connaissance qui soit réellement extensible. La seconde version est dite "creuse" par analogie avec les algorithmes de calcul scientifiques, car elle comporte des processeurs seulement pour les calculs utiles de l'algorithme.

3.2.2 CAO pour des circuits réguliers

Participants : Éric Gautrin, Laurent Perraudau, Oumarou Sié

Le but de nos recherches est de développer une chaîne complète de conception d'architectures régulières. Cette chaîne s'appuie sur deux outils existants développés au sein de l'équipe : l'environnement Alpha pour la synthèse de haut niveau et l'environnement Madmacs pour la synthèse physique grâce aux générateurs de dessins de masques qui y sont développés (thèse de O. Sié, [3]).

Les expérimentations menées sur la génération de dessins de masques de réseaux réguliers dans l'environnement Madmacs ont mis en évidence la nécessité de considérer la structure hiérarchique du réseau : processeur et réseau, pour obtenir des résultats satisfaisants. Pour la génération du processeur, il est important de contraindre le placement et le routage des cellules dont il est composé afin de préserver la régularité du niveau réseau. L'assemblage du réseau est alors ramené à un pavage de processeurs et à un routage régulier entre processeurs.

Nous avons évalué cette méthodologie en réalisant plusieurs générateurs de structures régulières linéaires ou bidimensionnels. Les différents exemples traités ont montré un gain de surface de 20% à 40% par rapport aux compilateurs de structures disponibles dans les chaînes de CAO actuelles.

Pour obtenir une chaîne complète de conception d'architectures régulières, la majeure partie de notre travail a porté sur l'interfaçage entre Alpha et Madmacs. Le problème essentiel est d'obtenir les informations topologiques (taille et forme du réseau, interconnectivité des processeurs, etc) afin d'automatiser la méthodologie de conception à deux niveaux de structures régulières et de générer les paramètres d'appel des générateurs. Cette approche a nécessité le développement de transformations dans l'environnement Alpha afin d'extraire puis d'analyser ces informations topologiques. De plus, il est nécessaire d'obtenir une description structurelle hiérarchique. Une première ébauche permet d'ores et déjà de traiter des exemples simples. Les travaux futurs devront nous permettre d'étendre notre champ d'applications aux réseaux hétérogènes (réseaux composés de différents types de processeurs) et aux réseaux à données autres que booléennes.

3.3 Études d'applications

Une part importante de l'activité du projet est consacrée à l'étude d'applications, dans le domaine du traitement vidéo, de la biologie moléculaire et de la compression de données. Les travaux menés en 1994 concernent les thèmes suivants :

- l'étude d'architectures pour le traitement de séquences biologiques ;
- l'étude d'architectures programmables temps-réel pour les traitements vidéo numériques ;

- l'étude d'architectures spécialisées pour l'estimation de mouvement dans des séquences d'images ;
- les architectures pour la compression de données, le cryptage et les codes correcteurs.

3.3.1 Architectures systoliques pour la comparaison de séquences

Participants : Pascale Guerdoux, Dominique Lavenier, Frédéric Raimbault, Charles Wagner

Une des principales méthodes de comparaison de séquences biologiques repose sur des algorithmes de programmation dynamique. La complexité de ces algorithmes conduit à des temps d'exécution prohibitifs sur des calculateurs séquentiels classiques, et incite à rechercher des solutions informatiques différentes, soit sur des *superordinateurs parallèles*, des *réseaux de calculateurs*, ou des *architectures spécialisées*. Samba (*Systolic Accelerator for Molecular Biology Application*) s'attaque au problème de la comparaison banque à banque de séquences biologiques [5]. Il concerne :

- l'aspect logiciel (parallélisation des traitements),
- l'aspect matériel (accélération des traitements).

Le premier aspect vise à *découper* l'application en tâches distinctes pouvant s'exécuter de manière indépendante, puis à gérer et à exploiter l'ensemble des résultats. Le second a pour but d'accélérer les parties excessivement coûteuses en calculs par adjonction de matériels spécialisés.

Ces deux aspects sont complémentaires : d'une part, la confrontation entre banques ne peut s'envisager que si la phase de comparaison (production d'alignements locaux, par exemple) s'effectue en un temps raisonnable et, d'autre part, la production rapide et en quantité de résultats issus de comparaisons entre séquences n'a de sens que si elle peut être traitée de manière performante.

L'objectif du projet est de mettre en place un environnement logiciel et matériel capable de traiter en un temps raisonnable (≤ 1 jour) des banques de séquences de quelques milliers d'éléments.

L'équipe Api a la charge de concevoir une machine spécialisée pour accélérer les traitements. Cette machine se compose d'un réseau systolique linéaire de plusieurs centaines de processeurs connecté à une station de travail via une interface spécialisée. Un circuit intégré comprenant quatre processeurs, réalisé au cours de cette année, est en cours de test.

3.3.2 Architectures programables temps-réel pour les traitements vidéo numériques

Participants : Ronan Barzic, François Charot, Gwendal Le Fol, Pascal Lemonnier, Charles Wagner

Les applications de traitement de signal 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 qui interagissent. La mise au point, l'émulation et l'évaluation de ces applications complexes nécessitent des architectures parallèles spécialisées.

En collaboration avec le CCETT, nous étudions l'utilisation des réseaux de processeurs SIMD pour la réalisation de simulateurs temps réel d'algorithmes de compression de séquences d'images animées. L'objectif de cette collaboration est la conception d'un processeur VLSI programmable permettant la réalisation de simulateurs temps réel adaptés aux schémas de compression de type MPEG, ainsi que son environnement de programmation.

Les travaux réalisés en 1994 ont porté sur la spécification de l'architecture de la machine parallèle de simulation temps-réel. Celle-ci se compose de modules (ou circuits VLSI) pouvant être organisés soit linéairement, soit en grille 2D. Un module, appelé Movie (*MOdule for VIdéo coding Experimentation*), peut être vu comme une machine systolique comportant un processeur d'entrées/sorties et un réseau linéaire de processeurs de calcul. Il dispose des mécanismes nécessaires à l'acquisition et à la restitution de la vidéo. La modélisation en langage VHDL de l'architecture est en cours.

3.3.3 Architectures spécialisées pour l'estimation de mouvement dans des séquences d'images

Participants : François Charot, Pascal Lemonnier

La parallélisation d'algorithmes d'estimation de mouvement pour le codage de séquences d'images fait l'objet d'une collaboration avec l'équipe Temis de Claude Labit. Les algorithmes considérés appartiennent à la famille des algorithmes "région-récurif". Ils sont basés sur des modèles affines (simplifiés ou complets) pour l'identification de mouvements non purement translationnels (paramètres de divergence, de rotation, etc.). Ces études visent à la spécification de circuits intégrés VLSI pour de tels algorithmes. À des fins de prototypage des algorithmes parallèles étudiés, des expérimentations sur des machines parallèles et en particulier sur la machine iWarp sont réalisées [22].

3.3.4 Architectures pour la compression de données, le cryptage, et les codes correcteurs

Participants : Patrice Quinton, Yannick Saouter

Certains traitements de fichiers se prêtent très bien à la systolisation. On peut citer ainsi la compression et le cryptage de données, ainsi que les techniques de codes correcteurs d'erreurs. Un survol de la littérature concernant les architectures de circuits dans les deux premiers domaines nous a permis de nous familiariser avec les principales techniques (compression Lempel-Ziv, Hoffman, algébrique, cryptage RSA, DES, IDEA ...). Indépendamment de cette étude, un contrat Cifre avec la société Nomaï sur la compression de données a été conclu, ce qui permettra d'amplifier ces recherches. En ce qui concerne les codes correcteurs, nous avons étudié les perspectives de sérialisation des calculs dans les corps finis. En effet, les corps de Galois sont très utilisés dans le sous-ensemble des codes par blocs. De nombreux travaux ont déjà été réalisés pour systoliser le calcul de la multiplication et de l'inversion dans les corps finis. Notre approche a consisté à sérialiser la multiplication dans les corps de Galois par des techniques parallèles non systoliques, afin d'optimiser les critères technologiques (fréquence maximale d'horloge et latence minimale). Ces multiplieurs doivent être intégrés dans des architectures de type systoliques comme le décodeur Reed-Solomon de Séroussi par exemple. Nous avons ainsi inventé une architecture de multiplieur sur le

corps de Galois à 256 éléments pour lequel nous avons soumis un brevet. Ce multiplieur a des applications assez immédiates pour le cryptage de données suivant l'algorithme de El Gamal.

4 Actions industrielles

4.1 Architecture à base de logique reconfigurable : Perle-1

Participants : Dominique Lavenier, Marc Vieillot

Cette action, menée dans le cadre d'un partenariat avec DEC-PRL, a comme objectif d'encourager le développement et le portage d'applications et d'outils utilisant les possibilités de calcul à grande vitesse des mémoires actives programmables (PAM) connectées à une station de travail.

4.2 Accélérateur matériel pour la comparaison de banques de séquences biologiques

Participants : Dominique Lavenier, Charles Wagner, Pascale Guerdoux

Ce projet, financé par le GIP GREG (Groupement de Recherches et d'études sur les Génomes), a pour but de concevoir une architecture spécifique – réseau systolique linéaire – permettant l'accélération significative (plusieurs ordres de grandeur par rapport à des machines programmables classiques) de calculs exécutés de manière intensive sur les bases de données génomiques.

4.3 Processeur VLSI SIMD pour le traitement vidéo

Participants : Ronan Barzic, François Charot, Gwendal Le Fol, Pascal Lemonnier, Charles Wagner

Cette action, menée dans le cadre d'une collaboration avec le CCETT, vise à la définition et à la conception d'un processeur VLSI programmable qui pourra servir de brique de base pour la simulation temps réel d'algorithmes de compression de séquences d'images et plus généralement de traitement de signaux vidéo-numériques.

4.4 Cairn, codesign d'applications irrégulières et régulières par niveaux

La proposition Cairn, soutenue par France-Télécom, collaboration entre les projet Api et EP-ATR, concerne l'utilisation conjointe des langages Signal et Alpha. Celle-ci a pour objectif d'assembler de façon cohérente et homogène – sous forme d'atelier – des outils permettant la spécification, la simulation et la réalisation matérielle et logicielle – autrement dit, le *codesign* d'une application. Ces travaux portent sur le thème de la dérivation d'architectures à partir de spécifications de haut niveau, d'une part pour des algorithmes réguliers et d'autre part pour des applications à contrôle complexe, en particulier temps réel.

4.5 Architectures pour la compression de données

Participants : Patrice Quinton, Yannick Saouter

Une convention Cifre a été établie entre la société Nomai et l'Irisa sur l'étude d'architectures pour la compression de données.

4.6 Encadrement de thèses en milieu industriel

Participants : Patrice Quinton

P. Quinton encadre la thèse d'Olivier Dugeon, au Cnet Lannion, sur la conception de serveurs spécialisés pour l'ATM, ainsi que la thèse de Claude Wicky (Faros – Lannion), sur la conception d'une architecture spécialisée pour la synthèse d'image. Cette thèse fait l'objet d'une convention Cifre entre Faros et l'Irisa.

5 Actions nationales et internationales

5.1 Actions nationales

Api participe au projet Paradigme du MESR, sur la compilation de langages à parallélisme de données, et au projet Asar du MESR, sur la synthèse d'architectures.

Le projet participe aux Groupements de Recherche du CNRS PRS (Parallélisme Réseaux et Systèmes), et ANM (Architectures Nouvelles de Machines).

P. Quinton est responsable du pôle Rennais du réseau doctoral en architecture des machines et des systèmes.

P. Quinton a fait partie du comité de programme de la conférence *Renpar 6*.

C. Wagner est membre du conseil d'administration de Jessica Ouest, ainsi que du comité de direction du CCMO (Centre Commun de Micro-électronique de l'Ouest).

Le projet entretient des relations avec l'UBO, le Lip et participe au séminaire régional Architecture (Enssat, ENST, Ireste, IUT de Vannes, UBO).

Dans le cadre de l'action génome, le projet entretient des relations suivies avec plusieurs laboratoires de biologie : Centre de Génétique Moléculaire de Gif sur Yvette, laboratoires de biologie cellulaire et de génétique moléculaire à Rennes.

5.2 Actions internationales

Api participe au projet Esprit *Basic Research Novel Algorithms for New Architectures II*, avec l'Imec, l'université de Louvain, l'université de Delft, l'Esat, et le Lip.

P. Quinton a fait partie des comités de programme des conférences *Asap'94* et *Parcella'94*.

P. Quinton est éditeur de la revue *Parallel Processing Letters*, et membre du comité de lecture de *Integration : the VLSI Journal*.

Le projet entretient des relations suivies avec les universités de Yale, de Delft, de Louvain, avec l'Imec, le Centre d'informatique de Sofia, l'université d'état d'Oregon, l'université de Dresde, et le Crim (Centre de Recherche en Informatique de Montréal). Un projet Franco-Québécois sur l'évaluation des logiciels de parallélisation a été commencé en 1994, avec le Crim.

Un programme Esprit Copernique avec les Universités de Sofia, de Novossibirsk, de Bratislava a été accepté par la communauté européenne. Ce projet, dont l'Irisa est contractant principal, implique Api et Aladin.

6 Diffusion des résultats

6.1 Enseignement

Des cours sur les architectures systoliques sont donnés par P. Quinton à l'ENST de Bretagne, à Supelec (Paris), à l'Enssat (Lannion), et à l'Ireste (Nantes).

F. Charot et P. Quinton sont responsables d'une option en DEA d'informatique (Rennes) sur les systèmes parallèles spécialisés, et d'un cours sur les méthodes de conception d'architectures en DIIC.

Des cours sur la conception de circuits VLSI sont donnés en DIIC par É. Gautrin, D. Lavenier et C. Wagner, ainsi qu'en DESS de micro-électronique par D. Lavenier.

Des travaux pratiques sont assurés en DEA d'électronique et en DIIC sur les outils de CAO par L. Perraudau.

Api a accueilli les stagiaires suivants : M. Le Boudier, P. Lemoënner, V. Messe, (DEA informatique), P. Guerdoux (institut Pasteur).

6.2 Participation à des conférences et séminaires

D. Lavenier a participé à l'école d'été du Parallélisme du CNRS à Lyon et a donné une présentation sur les architectures parallèles pour la comparaison des séquences biologiques.

P. Quinton a donné un cours sur les architectures spécialisées à l'école d'été des Jeunes Chercheurs en Architectures de Machines à Couiza (Aude), en juillet 1994.

6.3 Autres

Les 7 et 8 décembre 1993, l'équipe a organisé les journées des jeunes chercheurs en architectures de machines dans le cadre du réseau doctoral en architecture de machines et de systèmes.

Le 4 juin 1994, l'équipe a organisé une journée, sur le thème "Technologies futures des architectures et des circuits" réunissant industriels (CCETT, Cnet, Edixia, Faros, Thomson LER, Thomson CEB, Sorep) et laboratoires de recherche de l'Ouest. La mise en place d'un club régional en architecture a été décidée à l'issue de la journée.

7 Publications

Thèses

- [1] R. MCCONNELL, *Systèmes synchrones périodiques : modélisation et application*, thèse, université de Rennes 1, octobre 1994.
- [2] F. RAIMBAULT, *Étude et réalisation d'un environnement de simulation parallèle pour les algorithmes systoliques*, thèse, université de Rennes 1, janvier 1994.
- [3] O. SIÉ, *Génération automatique du dessin de masques de réseaux réguliers intégrés*, thèse, université de Rennes 1, décembre 1994.

Articles et chapitres de livre

- [4] R. ANDONOV, P. QUINTON, S. RAJOPADHYE, D. WILDE, «A shift register based systolic array for the general knapsack problem», *Parallel Processing Letters*, à paraître en 1995.
- [5] L. AUDOIRE, J. CODANI, D. LAVENIER, P. QUINTON, «Machines spécialisées pour la comparaison de séquences biologiques», *Technique et Science Informatiques (TSI)*, à paraître en janvier 1995.
- [6] F. CHAROT, P. FRISON, E. GAUTRIN, D. LAVENIER, P. QUINTON, C. WAGNER, «From equations to hardware. Towards the systematic mapping of algorithms onto parallel architectures», *International Journal of Pattern Recognition and Artificial Intelligence* 8, 2, 1994, p. 417–438.
- [7] F. CHAROT, «Architectures parallèles spécialisées pour l'image», *Technique et Science Informatiques (TSI)* 13, 3, 1994, p. 349–383.
- [8] D. LAVENIER, F. RAIMBAULT, P. FRISON, «I/O and computation overlap on SIMD systolic Arrays», *Journal of VLSI Signal Processing*, à paraître en 1995.
- [9] R. MCCONNELL, B. MENEZES, «Evaluation of task allocation in mesh-based multicomputers using genetic, neural, and traditional algorithms», *International Journal on Artificial Intelligence Tools* 3, 1, août 1994, p. 47–60.
- [10] E. MEMIN, F. HEITZ, F. CHAROT, «Efficient parallel non-linear multi-grid relaxation algorithms for low-level vision applications», *Journal of Parallel and Distributed Computing*, à paraître en 1995.

Communications à des congrès, colloques, etc.

- [11] R. ANDONOV, P. QUINTON, S. RAJOPADHYE, D. WILDE, «A shift register based implementation of the knapsack problem recurrences», *in : Parcelle'94*, Akademie Verlag, p. 207–214, 1994.
- [12] P. ASAR, «Towards a multi-formalism framework for architectural synthesis: the ASAR project», *in : Codes/CASHE'94*, IFIP WG 10.5, p. 25–32, Grenoble, France, septembre 1994.
- [13] J.-P. BANÂTRE, D. LAVENIER, M. VIEILLOT, «From high level programming model to FPGA machines», *in : FPGAs for Custom Computing Machines*, p. 119–124, Napa Valley, California, avril 1994.
- [14] M. BELHADJ, A. KERIHUEL, R. MCCONNELL, «Une comparaison des approches pour la conception des circuits synchrones et asynchrones», *in : actes des Journées des jeunes chercheurs en architecture de machines*, Rennes, France, décembre 1993.
- [15] J. CHAMPEAU, E. GAUTRIN, L. LE PAPE, L. PERRAUDEAU, B. POTTIER, S. RUBINI, «Flexible parallel FPGA-based architectures with ArMen», *in : Twenty-seventh Hawaii International Conference of System Sciences*, p. 105–113, Hawaii, janvier 1994.
- [16] C. DEZAN, P. QUINTON, «Verification of regular architectures using Alpha : a case study», *in : ASAP'94*, IEEE Computer Society Press, p. 164–176, San Francisco, août 1994.
- [17] E. GAUTRIN, O. SIÉ, «A new cell placement algorithm for optimal linear layout», *in : 2^e Colloque Africain sur la Recherche en Informatique - CARI'94*, Ouagadougou, octobre 1994.
- [18] A. KERIHUEL, R. MCCONNELL, S. RAJOPADHYE, «Des graphes de flots de données synchrones pour le VLSI», *in : Actes des 6èmes rencontres francophones du parallélisme*, ENS Lyon, France, juin 1994.
- [19] A. KERIHUEL, R. MCCONNELL, S. RAJOPADHYE, «VSDF: synchronous data flow for VLSI», *in : Proceedings of the 37th Midwest Symposium on Circuits and Systems*, Lafayette, Louisiana, août 1994.
- [20] D. LAVENIER, R. MCCONNELL, «From behavioral to RTL models: an approach», *in : Proceedings of the 5th IEEE International Workshop on Rapid System Prototyping*, Grenoble, France, juin 1994.
- [21] D. LAVENIER, «Architecture Parallèle pour la comparaison de séquences biologiques», *in : école d'été du CNRS sur le parallélisme*, Lyon, juillet 1994.
- [22] G. LE FOL, P. LEMONNIER, «iWarp comme machine de prototypage», *in : Actes des 6èmes rencontres francophones du parallélisme*, p. 249–252, ENS Lyon, France, juin 1994.

- [23] E. MEMIN, F. HEITZ, F. CHAROT, «Efficient parallel multigrid relaxation algorithms for markov random field-based low-level vision applications», *in: IEEE Conference on Computer Vision and Pattern Recognition*, p. 644–648, Seattle USA, juin 1994.
- [24] P. QUINTON, «Systolic arrays: why and how?», *in: Parcella' 94*, Akademie Verlag, p. 39–50, 1994. Conférence invitée.
- [25] M. VIEILLOT, «À chaque programme sa machine», *in: Actes des 6èmes rencontres francophones du parallélisme*, p. 263–266, ENS Lyon, France, juin 1994.
- [26] D. K. WILDE, O. SIÉ, «Regular Array Synthesis using Alpha», *in: International Conference on Application Specific Array Processors - ASAP'94*, p. 200–211, San Francisco, août 1994.

Rapports de recherche et publications internes

- [27] R. ANDONOV, S. RAJOPADHYE, «A sparse knapsack algo-tech-cut and its synthesis», *publication interne n° 801*, Irisa, février 1994.
- [28] R. ANDONOV, S. RAJOPADHYE, «An optimal algo-tech-cut for the knapsack problem», *publication interne n° 791*, Irisa, janvier 1994.
- [29] R. ANDONOV, S. RAJOPADHYE, «Optimal tiling», *publication interne n° 792*, Irisa, janvier 1994.
- [30] L. AUDOIRE, J. CODANI, D. LAVENIER, P. QUINTON, «Machines spécialisées pour la comparaison de séquences biologiques», *publication interne n° 795*, Irisa, janvier 1994, paru aussi comme rapport de recherche Inria n° 2251.
- [31] J.-P. BANÂTRE, D. LAVENIER, M. VIEILLOT, «From high level programming model to FPGA machines», *publication interne n° 810*, Irisa, janvier 1994, paru aussi comme rapport de recherche Inria n° 2240.
- [32] C. DEZAN, P. QUINTON, «Verification of regular architectures using Alpha : a case study», *publication interne n° 823*, Irisa, mai 1994.
- [33] A. KERIHUEL, R. MCCONNELL, S. RAJOPADHYE, «VSDF: synchronous data flow for VLSI», *publication interne n° 843*, Irisa, juin 1994.
- [34] D. LAVENIER, R. MCCONNELL, «A Component Model for Synchronous VLSI System Design», *publication interne n° 822*, Irisa, mai 1994, paru aussi comme rapport de recherche Inria n° 2285.
- [35] D. LAVENIER, F. RAIMBAULT, P. FRISON, «I/O and computation overlap on SIMD systolic Arrays», *publication interne n° 770*, Irisa, novembre 1993.

- [36] H. LE VERGE, Y. SAOUTER, «New results on calculability of systems of recurrence equations», *publication interne n° 380*, LaBRI, septembre 1993, Soumis à *Foundations of Computer Science*.
- [37] H. LE VERGE, V. VAN DONGEN, D. WILDE, «Loop nest synthesis using the polyhedral library», *publication interne n° 830*, Irisa, mai 1994.
- [38] E. MEMIN, F. HEITZ, F. CHAROT, «Efficient parallel non-linear multi-grid relaxation algorithms for low-level vision applications», *publication interne n° 790*, Irisa, janvier 1994, paru aussi comme rapport de recherche Irisa n°2184, avril 1994.
- [39] P. QUINTON, S. RAJOPADHYE, D. WILDE, «Using static analysis to derive imperative code from Alpha», *publication interne n° 828*, Irisa, mai 1994.
- [40] Y. SAOUTER, «Optimization of number of processors in VLSI arrays», *publication interne n° 850*, Irisa, septembre 1994, paru aussi comme rapport de recherche Inria n°2333, soumis à *Intégration*.
- [41] D. WILDE, S. RAJOPADHYE, «An inductive constructive method for computation of the face lattice of a polyhedron», *publication interne n° 786*, Irisa, décembre 1993.
- [42] D. WILDE, O. SIÉ, «Regular array synthesis using Alpha», *publication interne n° 829*, Irisa, mai 1994.
- [43] D. WILDE, «A library for doing polyhedral operations», *publication interne n° 785*, Irisa, décembre 1993.
- [44] D. WILDE, «Le langage Alpha», *publication interne n° 827*, Irisa, mai 1994.

Table des matières

1	Composition de l'équipe	1
2	Présentation générale et objectifs	3
3	Actions de recherche	4
3.1	Outils pour la conception et la simulation d'architectures spécialisées	5
3.1.1	Prototypage de systèmes matériels à l'aide de langages flots de données synchrones	6
3.1.2	Environnement de programmation pour algorithmes systoliques	7
3.1.3	Génération de code optimisé pour des processeurs spécialisés programmables	8
3.1.4	Mise en œuvre de Gamma sur Perle	8
3.2	Synthèse d'architectures parallèles régulières	9
3.2.1	Alpha et la synthèse d'architectures régulières . . .	10
3.2.2	CAO pour des circuits réguliers	11
3.3	Études d'applications	12
3.3.1	Architectures systoliques pour la comparaison de séquences	13
3.3.2	Architectures programables temps-réel pour les traitements vidéonumériques	14
3.3.3	Architectures spécialisées pour l'estimation de mouvement dans des séquences d'images	15
3.3.4	Architectures pour la compression de données, le cryptage, et les codes correcteurs	15
4	Actions industrielles	16
4.1	Architecture à base de logique reconfigurable : Perle-1 . .	16
4.2	Accélérateur matériel pour la comparaison de banques de séquences biologiques	16
4.3	Processeur VLSI SIMD pour le traitement vidéo	16

4.4	Cairn, codesign d'applications irrégulières et régulières par niveaux	17
4.5	Architectures pour la compression de données	17
4.6	Encadrement de thèses en milieu industriel	17
5	Actions nationales et internationales	17
5.1	Actions nationales	17
5.2	Actions internationales	18
6	Diffusion des résultats	19
6.1	Enseignement	19
6.2	Participation à des conférences et séminaires	19
6.3	Autres	19
7	Publications	20