

Équipe R2D2

*Exploration, estimation, prototypage pour
la conception de systèmes matériels sur des
plates-formes sur silicium reconfigurable*

Rennes

THÈME 1A



*R*apport
*d'*Activité

2002

Table des matières

1. Composition de l'équipe	1
2. Présentation et objectifs généraux	2
2.1. Introduction	2
2.2. Nouvelles architectures et technologies	2
2.3. Synthèse d'accélérateurs parallèles dédiés	3
2.4. Exploration, estimation, prototypage pour la conception de systèmes sur silicium	4
3. Fondements scientifiques	5
3.1. Panorama	5
3.2. Parallélisation de programmes réguliers et synthèse d'accélérateurs matériels dédiés	5
3.3. Modélisation de processeurs et compilateurs flexibles	5
3.4. Implantation d'algorithmes spécifiés en virgule flottante dans des architectures virgule fixe	6
3.5. Nouvelles architectures reconfigurables	7
4. Domaines d'application	8
4.1. Panorama	8
4.2. Télécommunications mobiles	8
4.3. Filtrage adaptatif	8
4.4. Algorithme de chiffrement AES	8
5. Logiciels	9
5.1. Panorama	9
5.2. PolyLib	9
5.3. MMAAlpha	9
5.4. BSS	10
6. Résultats nouveaux	11
6.1. Nouvelles architectures et technologies	11
6.1.1. Architecture faible consommation	11
6.1.2. Architecture reconfigurable Dart	11
6.1.3. Architectures et circuits en logique à valeurs multiples (MVL)	12
6.2. Synthèse d'accélérateurs parallèles dédiés	12
6.2.1. Techniques de partitionnement pour l'implantation de réseaux de processeurs sur FPGA	13
6.2.2. Génération de code VHDL	13
6.2.3. Génération automatique d'interfaces	13
6.2.4. Utilisation de méthodes formelles pour la détermination de propriétés de systèmes Alpha	13
6.2.5. Méthodologie pour la détermination de la largeur des chemins de données	14
6.2.6. Travaux sur la sérialisation	14
6.3. Exploration, estimation, prototypage pour la conception de systèmes sur silicium	14
6.3.1. Langage de description d'architectures	14
6.3.2. Estimation de performances logicielles	14
6.3.3. Méthodologie d'implantation d'algorithmes spécifiés en virgule flottante dans des architectures virgule fixe	15
6.3.4. Environnement logiciel de l'architecture Dart	15
6.4. Etude d'applications	16
6.4.1. Prototypage d'applications de télécommunications mobiles	16
6.4.2. Codage d'images et de vidéo	16
6.4.3. Indexation d'images	16
7. Contrats industriels	17
7.1. Hades : PEA sécurisation des réseaux hauts débits (2002-2003)	17
7.2. IST Ozone (2002-2004)	17

7.3.	PHRASE : reconfigurabilité et processeurs VLIW dans les architectures parallèles hétérogènes	17
7.4.	Architectures basée sur la logique à valeurs multiples pour applications en télécommunications (2001-2004)	18
8.	Actions régionales, nationales et internationales	18
8.1.	Actions nationales	18
8.2.	Relations bilatérales internationales	18
8.2.1.	Europe	18
8.2.2.	Afrique	18
8.2.3.	Amérique du Nord	19
9.	Diffusion des résultats	19
9.1.	Animation de la communauté scientifique	19
9.2.	Enseignement universitaire et responsabilités	19
9.3.	Accueil d'étudiants stagiaires	20
10.	Bibliographie	20

1. Composition de l'équipe

R2D2 est une action créée auprès de l'Unité de Recherche de Rennes à la date du 15 mai 2002. L'action regroupe des personnels de l'unité de recherche de Rennes (membres de l'ancien projet COSI¹) et les membres du groupe signal-architecture du laboratoire Lasti de l'Enssat à Lannion.

Responsables scientifiques

François Charot [CR Inria]

Olivier Sentieys [Professeur, université de Rennes 1, Enssat, en détachement CR Inria jusqu'au 31/08/2002]

Assistante de projet

Maryse Auffray [AA Inria, jusqu'au 31/10/2002]

Lydie Letort [TR Inria, à compter du 1/11/2002]

Personnel Inria

Sorin Olaru [Ingénieur associé Inria jusqu'au 25/10/2002]

Personnel CNRS

Sanjay Rajopadhye [CR CNRS, en disponibilité à l'université d'État du Colorado (Fort Collins, USA)]

Charles Wagner [IR CNRS (Atelier)]

Personnel Université

Michel Aline [ATER, université de Rennes 1, Enssat]

Daniel Chillet [Maître de conférences, université de Rennes 1, Enssat]

Steven Derrien [1/2 ATER, université de Rennes 1, Ifsic]

Hélène Dubois [Maître de conférences, université de Rennes 1, Enssat]

Anne-Claire Guillou [ATER, université de Rennes 1, Ifsic, à compter du 1/09/2002]

Michel Guitton [Maître de conférences, université de Rennes 1, Enssat]

Daniel Menard [IR Enssat en détachement sur un poste de 1/2 ATER]

Laurent Perraudeau [Maître de conférences, université de Rennes 1, Ifsic]

Sébastien Pillement [Maître de conférences, université de Rennes 1, Iut Lannion]

Patrice Quinton [Professeur, université de Rennes 1, Ifsic]

Taoufik Saidi [IR, université de Rennes 1, Enssat]

David Saillé [1/2 ATER, université de Rennes 1, Enssat, jusqu'au 30/08/02]

Pascal Scalart [PAST, université de Rennes 1, Enssat, France Telecom R&D]

Chercheurs doctorants

Imène Benkermi [bourse université, Enssat]

Stéphane Chevobbe [bourse CEA à compter du 1/10/02, Enssat]

Alexandre Buisson [contrat FT R&D Rennes jusqu'au 31/10/02, Enssat]

Raphaël David [bourse MENRT, Enssat]

Ekoué Kinvi-Boh [bourse MENRT, Enssat]

Anne-Claire Guillou [bourse Inria jusqu'au 31/08/2002, Irisa]

Gautam Gupta [bourse CIES, co-tutelle avec l'université d'État du Colorado, Irisa]

Nicolas Hervé [bourse université à compter du 1/11/2002, Enssat]

Ludovic L'Hours [bourse MENRT à compter du 01/10/2002, Irisa]

Katell Morin-Allory [bourse MENRT Irisa, a rejoint le projet Lande au 1/07/2002]

Madeleine Nyamsi [bourse Inria, à compter du 15/03/02, Irisa]

Jean-Marc Philippe [bourse cofinancée Région à compter du 1/10/2002, Enssat]

Collaborateur extérieur

David Cachera [maître de conférences ENS-Cachan, a rejoint le projet Lande au 1/07/2002]

¹ Le projet COSI a été arrêté au 30 juin 2002, comme prévu par la recommandation du 10 octobre 2000 de la commission d'évaluation (<http://www.inria.fr/DR:/dg/viescientifique/decisions/000972noteeval1a.pdf>)

2. Présentation et objectifs généraux

2.1. Introduction

La problématique générale de R2D2 (*Reconfigurable and Retargetable Digital Devices*) est la conception de systèmes matériels spécialisés sur silicium. Nous abordons cette conception sous l'angle de l'exploration, l'estimation et le prototypage pour des plates-formes matérielles programmables et reconfigurables.

On appelle plate-forme matérielle (au sens *platform-based design* [41]) un schéma d'architecture intégrée, commun à de nombreux algorithmes ou applications appartenant à un même domaine applicatif. Cette structure de circuit intégré comporte en général un ensemble de composants programmables (cœurs de processeurs, blocs de logique reconfigurable), des mémoires et éventuellement quelques composants spécialisés. Cette notion est la réponse que les constructeurs de systèmes enfouis donnent à la difficulté croissante qu'ils rencontrent à réaliser leurs applications.

On peut imaginer qu'à l'avenir, la plupart des circuits intégrés nécessaires à la réalisation d'un système spécialisé complexe seront dérivés d'une plate-forme matérielle existante donnée. Cette approche de conception est une alternative à une conception réalisée par assemblage d'une collection de composants virtuels (blocs IP, *Intellectual Property*) conçus séparément.

Il ne fait par ailleurs aucun doute que ces plates-formes seront très fortement parallèles, hétérogènes, programmables, et reconfigurables. Le parallélisme est la seule façon d'atteindre les performances requises par les futures applications. L'hétérogénéité résulte du constat que dans un système, il y a généralement plusieurs sous-systèmes, bien identifiés, caractérisés par des besoins en calcul bien différenciés. La programmabilité évite de figer les fonctionnalités et permet ainsi d'explicitier le comportement du système par un programme. La reconfigurabilité combine la rapidité de solutions spécialisées et la souplesse de composants programmables classiques.

L'objectif scientifique de l'action R2D2 est caractérisé par les éléments suivants.

- Un domaine applicatif privilégié, celui des télécommunications mobiles de troisième et quatrième générations. Compte tenu de la complexité croissante des applications dans des domaines comme ceux du traitement du signal et de l'image, de l'évolution de l'offre de services en télécommunications et des technologies ainsi que des nouvelles contraintes liées à la mobilité des systèmes (flexibilité, performances, faible consommation), ce domaine applicatif est propice à la mise en valeur de nos résultats.
- Un cadre méthodologique. Nous cherchons à tirer parti de différentes méthodes (synthèse de très haut niveau, synthèse d'architecture, compilation flexible, génération de code virgule fixe, etc.), contribuant chacune avec ses spécificités, à la conception d'une partie d'un système matériel spécialisé. Les modèles et les techniques sous-jacentes permettent la mise en place d'estimateurs, contribuant ainsi aux choix de mise en œuvre, à une connaissance précise des performances du système, de sa complexité et de sa consommation.

Les recherches menées au sein de l'action R2D2 visent à faciliter la conception de systèmes matériels reconfigurables sur silicium, en proposant des modèles d'architectures et des méthodologies de conception associées qui privilégient l'adéquation entre les algorithmes des applications traitées et les architectures matérielles supportant l'implémentation.

2.2. Nouvelles architectures et technologies

Mots clés : *architecture reconfigurable, grain de calcul, logique multivalente, haute-performance, faible consommation.*

D'ici la fin de la décennie, la technologie des circuits intégrés devrait permettre d'intégrer un milliard de transistors sur une puce de silicium, au lieu des quelques dizaines de millions aujourd'hui comme l'illustrent

les documents publiés par la SIA² (*Semiconductors Industry Association*). Les systèmes matériels qui seront la base des futurs équipements seront miniaturisés - on parle maintenant couramment de système matériel mono-puce³ - tout en mélangeant des architectures qui pourront être hétérogènes et inclure des accélérateurs matériels dédiés.

Même si les outils de CAO électronique et les méthodologies de conception associées ont beaucoup progressé au cours des dernières années, la conception de nouveaux circuits intégrés n'en est pas pour autant plus facile aujourd'hui. Au contraire, la distance entre les capacités offertes par la technologie et le potentiel des outils de conception actuels - le fameux *fossé technologique*, - n'a jamais été aussi grande. Un changement assez fondamental dans la façon de concevoir les circuits semble se dessiner.

Cette évolution de la technologie a un impact sur les architectures des circuits intégrés. Au fil des ans, une migration des architectures est constatée : des circuits Asic « purs » vers les systèmes mono-puces, et dans un futur proche vers les plates-formes programmables reconfigurables.

- Les Asic « purs » ont été prédominants entre 1980 et 1995, et ne sont désormais utilisés que comme des blocs particuliers dans des systèmes hétérogènes plus complexes.
- Les premiers systèmes mono-puces (SoC) ont vu le jour dans les années 1995. Tirant parti de la densité croissante des puces, un système mono-puce complexe intègre couramment un ou plusieurs cœurs de processeurs (processeur à usage général ou de traitement de signal), des blocs de mémoire (RAM, ROM, mémoire flash, EPROM, etc.), ainsi que les nombreuses interfaces utiles au bon fonctionnement du système. Ils combinent des composantes matérielles et logicielles. Leur conception repose sur l'utilisation d'outils de synthèse, de placement-routage, de plan de masse et d'assemblage au niveau du circuit, et de bibliothèques de composants réutilisables.
- Dans un proche avenir, les systèmes mono-puces vont évoluer vers des plates-formes, ces schémas d'architectures intégrées, communs à un ensemble d'algorithmes ou applications appartenant à un même domaine applicatif. Les outils et méthodologies de conception doivent donc permettre de concevoir une architecture spécialisée à partir de cette architecture de base [51]. Les plates-formes permettront de répondre aux besoins d'un spectre plus large d'applications, au prix d'une réduction de la variété des circuits conçus.

Les actions de recherche que nous menons portent sur l'étude de nouvelles organisations de structures reconfigurables offrant la rapidité de solutions spécialisées avec la souplesse de composants programmables classiques, en lien avec des contextes applicatifs issus du domaine des télécommunications. Ces nouvelles organisations constituent l'un des composants des plates-formes reconfigurables. Les travaux portent en particulier sur la définition d'un modèle architectural, appelé Dart associant haute performance et faible consommation. Cette architecture cible les terminaux multimédia portables : elle doit être capable de traiter des applications très différentes en termes de granularité, de motifs de calcul ou de contraintes temporelles.

Nos travaux portent par ailleurs sur l'étude d'architectures et de circuits en logique à valeurs multiples (MVL pour *Multiple Valued Logic*). De nos jours, les systèmes numériques sont exclusivement basés sur une représentation binaire des nombres et des calculs. Il a été démontré que l'utilisation d'un nombre supérieur d'états logiques peut permettre d'optimiser le rendement des traitements, des interconnexions et des éléments de mémoire [35]. Ces techniques permettent en particulier d'améliorer la bande passante lors de la transmission ou le stockage de données, de réaliser des fonctions complexes en un temps inférieur et avec une consommation moindre que la logique binaire ou d'utiliser des fonctions de cryptage complexes.

2.3. Synthèse d'accélérateurs parallèles dédiés

Mots clés : *synthèse d'architecture de haut niveau, CAO, architecture parallèle, circuit intégré, méthodologie de conception.*

²<http://public.itrs.net/Files/2001ITRS/>

³SoC, *System on a Chip* en anglais.

Bien que l'architecture des circuits intégrés évolue vers des solutions de plus en plus programmables et reconfigurables, il n'en reste pas moins vrai que les futurs systèmes sur silicium continueront à intégrer des composants matériels spécialisés. La conception de tels composants passe par l'utilisation de techniques de synthèse. En conception de circuits, on appelle synthèse le processus qui permet de passer d'une description comportementale à une description architecturale qui la met en œuvre. Les outils de conception actuels incluent généralement différents logiciels de synthèse. Au cours des dernières années la tendance des outils est d'élever le niveau d'abstraction des spécifications. On appelle traditionnellement synthèse de très haut niveau la synthèse qui permet l'obtention d'une architecture à partir d'un programme.

Les méthodologies étudiées s'appuient sur le modèle polyédrique, qui est bien adapté à l'expression des parties de calcul des applications et qui permet l'expression et la manipulation de systèmes d'équations récurrentes. Ce modèle est à la base des principaux formalismes développés pour la synthèse d'architectures régulières, formalismes dont fait partie le langage Alpha, développé successivement par les projets Api et Così.

Les actions de recherche que nous menons visent à la conception d'accélérateurs matériels parallèles dédiés pour les parties de calcul intensif des applications. Ces accélérateurs matériels constituent un des composants des plates-formes reconfigurables. Les méthodologies étudiées prennent en compte la conception des mécanismes d'interface nécessaire au contrôle, à l'initialisation et à l'alimentation efficace de l'accélérateur matériel. Ces travaux font l'objet d'une coopération étroite avec l'action CompSys du LIP (ENS Lyon).

2.4. Exploration, estimation, prototypage pour la conception de systèmes sur silicium

Mots clés : *synthèse d'architecture, compilation flexible, modélisation d'architecture, conception d'ASIP, arithmétique virgule fixe.*

La mise en œuvre d'une application sur une telle plate-forme reconfigurable nécessite la mise en place d'un ensemble de techniques variées (synthèse d'architecture, compilation flexible, génération de code virgule fixe, profilage, etc.) qui, par raffinements successifs, contribuent aux choix d'implémentation des différentes parties de l'application sur les composants de la plate-forme. Les actions de recherche que nous menons visent à mettre en place des méthodologies permettant la mise en œuvre des différentes parties de l'application sur les différents composants de la plate-forme.

L'implantation efficace d'un traitement ou algorithme sur un processeur spécialisé, comme par exemple un processeur de traitement de signal ou un Asip (*Application Specific Instruction-set Processor*), nécessite pour des raisons liées aux contraintes de coût, de consommation ou de surface de silicium, l'utilisation de l'arithmétique virgule fixe, alors que les algorithmes sont habituellement spécifiés en arithmétique virgule flottante. Nous développons une méthodologie pour transformer une spécification algorithmique en virgule flottante en une spécification en virgule fixe. La méthodologie doit déterminer le codage optimal des données, permettant de maximiser la précision et de minimiser le temps d'exécution et la taille du code.

La modélisation des systèmes matériels est un aspect clé du processus d'exploration de l'espace de conception d'architectures. Nos études concernent l'étude de langages de description d'architectures. Ceux-ci jouent un rôle important dans le développement des architectures, leur dimensionnement vis-à-vis des contraintes des applications cibles, et le support des environnements de développement logiciel associés (compilateur, simulateur, outils de conception du processeur, etc.).

Pour permettre la mise en place de ces méthodologies et la réalisation d'expérimentations, nous développons une infrastructure de compilation flexible, Calife, permettant le prototypage de compilateurs pour des processeurs spécialisés de type Asip, modélisés à l'aide du langage de description Armor.

3. Fondements scientifiques

3.1. Panorama

Les activités de recherche de l'action R2D2 s'appuient sur des bases issues de deux communautés scientifiques dont les compétences sont tout à fait complémentaires pour la conception de systèmes sur silicium : la première concerne les méthodes et outils de conception d'architectures spécialisées et la seconde le traitement du signal et les architectures de circuits dédiés.

Nous présentons ici brièvement quelques fondements de nos recherches : les principes et défis liés à la parallélisation de programmes réguliers et à la conception d'accélérateurs matériels dédiés, un aperçu des techniques de modélisation de processeurs spécialisés, le problème de l'implantation d'algorithmes spécifiés en virgule flottante dans les architectures virgule fixe, ainsi que les défis liés à l'étude de nouvelles architectures reconfigurables.

3.2. Parallélisation de programmes réguliers et synthèse d'accélérateurs matériels dédiés

Mots clés : *synthèse d'architecture de haut niveau, CAO, architecture parallèle, circuit intégré, méthodologie de conception.*

La synthèse de circuits se fait aujourd'hui à partir de spécifications de plus en plus haut niveau. La spécification de programmes effectuant des calculs réguliers sous forme d'équations récurrentes permet des analyses statiques puissantes et des transformations de programmes pour la dérivation d'architectures régulières [7].

Le fondement de nos recherches est le modèle polyédrique, qui est bien adapté à l'expression des parties de calcul des applications et qui permet l'expression et la manipulation de systèmes d'équations récurrentes.

Il existe de nombreux prototypes d'environnements académiques pour la synthèse automatique d'architectures spécialisées à partir de spécification de haut niveau : par exemple, Diastol, Presage, Hifi, Cathedral, Sade, PEI et MMAAlpha. On voit par ailleurs apparaître sur le marché des outils réalisant une synthèse de haut niveau, notamment à partir de C : outils basés sur SystemC⁴ comme *CoCentric SystemC Compiler*⁵ de Synopsys, *A|RT Builder*⁶ de Adelante Technologies/Frontier Design, outils basés sur C et des extensions comme *Celoxica DK1 Design Suite* de Celoxica.

Peu d'outils reposent sur une véritable parallélisation mais de nombreux projets de recherche explorent cette voie : Flex⁷ et Raw⁸ au MIT, Piperench à Carnegie-Mellon, Garp⁹ à Berkeley, Pico [52] à HPLabs Palo Alto, Compaan¹⁰ à Leiden.

Alpha [2] et MMAAlpha, développés dans le projet Cosis, ont évolué à partir de Diastol et constituent aujourd'hui un environnement pratique pour la manipulation d'équations récurrentes affines et la synthèse (dite de *très haut niveau*) d'accélérateurs matériels dédiés.

3.3. Modélisation de processeurs et compilateurs flexibles

Mots clés : *langage de description d'architecture, ASIP, processeur spécialisé, compilation recyclable, compilation flexible.*

Les langages de description de matériel comme VHDL ou Verilog sont largement utilisés pour modéliser et simuler des processeurs, mais principalement dans le but de concevoir le matériel. La conception de systèmes sur silicium (SoC) nécessite des méthodologies et des outils pour l'exploration de l'espace de conception

⁴<http://www.systemc.org>

⁵http://www.synopsys.com/products/cocentric_studio/

⁶<http://www.adelantetech.com/en/html/algemeen/Products/ARTproducts/Builder/Builder.asp>

⁷<http://flex-compiler.lcs.mit.edu>

⁸<http://cag.lcs.mit.edu/raw>

⁹<http://brass.cs.berkeley.edu/garp.html>

¹⁰<http://www.liacs.nl/~cserc/compaan/index.html>

des architectures. Cette exploration passe par l'utilisation de langages de description d'architectures (LDA), adaptés à la spécification des modèles d'architecture des SoC. Ils ont pour rôle d'une part la validation des architectures de SoC très tôt dans le processus de conception, et d'autre part la génération automatique des outils de développement logiciel nécessaires à la conception logicielle et matérielle de l'architecture.

La plupart des langages de description d'architectures existants vise à la spécification d'architectures de processeurs, en privilégiant soit la synthèse, soit la génération de compilateurs, soit la génération de simulateurs, mais très rarement l'ensemble. Aucun des langages existants n'est véritablement orienté vers l'exploration architecturale.

Dans la catégorie des langages de description d'architectures principalement orientés vers la synthèse matérielle du processeur, on peut citer Mimola, développé à l'université de Dortmund, et utilisé pour décrire des machines cibles dans les compilateurs MSSQ et Record [43]. Mimola est très proche d'un langage de description de matériel comme VHDL ou Verilog. Une description Mimola peut être employée pour la synthèse, la simulation, et la génération de code, après extraction du jeu d'instructions de la description.

En ce qui concerne les langages de description d'architectures principalement orientés vers la compilation, on peut citer nML, conçu à l'université de Berlin, ISDL proposé par le MIT, MDES développé à l'université d'Illinois, Expression développé à l'université de Californie Irvine.

En ce qui concerne les langages de description d'architectures principalement orientés vers la simulation, on peut citer LISA [47], développé à l'université d'Aix la Chapelle. LISA permet la génération de simulateurs précis au niveau du cycle pour des processeurs de type DSP. A la fois la structure et le comportement peuvent être modélisés.

Les langages de description d'architectures existants peuvent être classés selon le niveau de modélisation autorisé : comportemental ou structurel. Un langage comme Mimola est de niveau structurel, les langages comme nML et ISDL sont de niveau comportemental. LISA, Expression et MDES mélangent les deux niveaux de modélisation.

Il n'y a pas de standard en matière de langage de description d'architectures, comme l'atteste ce panorama. Le langage Armor développé dans le projet Cosi, de par sa modularité et sa concision nous semble être adapté à la modélisation d'architectures complexes de systèmes matériels, tant vis-à-vis de l'exploration architecturale que de la génération automatique d'outils de développement.

3.4. Implantation d'algorithmes spécifiés en virgule flottante dans des architectures virgule fixe

Mots clés : *méthodologie, arithmétique virgule fixe, codage des données, précision.*

Les algorithmes, tels que généralement proposés par leurs concepteurs et les comités de standardisation en charge de leur diffusion, manipulent couramment des données en virgule flottante. Leur implantation au sein de systèmes sur silicium supportant une arithmétique en virgule fixe nécessite de réaliser une conversion de la description de l'algorithme en virgule flottante en une spécification en virgule fixe afin de satisfaire les contraintes de coût, de consommation et d'encombrement exigées par les applications. Cette conversion est une tâche fastidieuse, longue et source d'erreurs si elle est réalisée manuellement. En effet, certaines expérimentations [37] ont montré que le temps consacré à cette phase de conversion en virgule fixe est relativement important, la conversion manuelle pouvant représenter jusqu'à 30% du temps global nécessaire à l'implantation de l'algorithme. Notons par ailleurs que la réduction du temps de mise sur le marché des applications exige l'utilisation d'outils de développement de haut niveau, permettant d'automatiser certaines tâches.

Les méthodologies de codage automatique des données en virgule fixe existantes [42][53] réalisent une transformation de la représentation des données en virgule flottante en une représentation en virgule fixe, sans prendre en considération l'architecture du processeur cible. Cependant l'analyse de l'influence de l'architecture sur la précision des calculs et des différentes phases de la génération de code [25] montre la nécessité de tenir compte des caractéristiques de l'architecture et de coupler les processus de codage et

génération de code pour obtenir une implantation de qualité en termes de précision des calculs et de temps d'exécution.

L'optimisation du codage des données doit être réalisée sous contrainte de précision, et il est donc nécessaire de déterminer le rapport signal à bruit de quantification en sortie de l'application. Les méthodes de détermination du RSBQ [40] sont généralement basées sur la simulation. Mais dans le cadre de l'optimisation du codage des données ces méthodes utilisent un processus itératif conduisant à des temps d'optimisation élevés. L'étude de méthode analytique de détermination du rapport signal à bruit de quantification est une voie que nous étudions.

3.5. Nouvelles architectures reconfigurables

Mots clés : *architecture reconfigurable, grain de calcul, faible consommation, haute-performance.*

Ces dernières années ont vu l'émergence de nouvelles architectures, qualifiées de reconfigurables, qui se présentent comme une nouvelle alternative au traditionnel compromis performance/flexibilité, conditionnant le choix entre solution purement matérielle (Asic) ou purement logicielle (processeur programmable). Néanmoins, malgré le grand nombre de projets sur les architectures reconfigurables [38], aucun d'entre eux n'ambitionne d'associer les trois principales contraintes inhérentes aux télécommunications de prochaine génération que sont la haute performance, la faible consommation et la flexibilité.

A titre d'exemple, le projet Pleiades [48] est une plate-forme architecturale supportant plusieurs grains de calculs, - les traitements logiques sont traités aussi efficacement que les traitements arithmétiques,- conçue afin de consommer un minimum d'énergie quel que soit le niveau de performance requis. Cette plate-forme ne permet cependant pas de supporter l'ensemble des contraintes précédemment évoquées en raison du caractère statique de sa reconfiguration qui la limite à certains domaines applicatifs, le codage de parole ayant été le support d'étude.

La reconfiguration dynamique proposée par le processeur Chameleon¹¹ se traduit par une flexibilité suffisante pour couvrir les besoins inhérents à un domaine applicatif, tout en autorisant la gestion d'un nombre de ressources de calcul permettant le traitement d'une chaîne complète de communication de troisième génération. Malgré cela, le manque de maîtrise de sa consommation interdit son utilisation au sein d'un terminal multimédia portable.

Outre ces deux exemples, de nombreuses architectures reconfigurables se basent sur des circuits de type FPGA et la plupart d'entre eux, tels GARP [39], NAPA [49], CHIMAERA [50], intègrent un processeur programmable traditionnel chargé du séquençage des traitements sur le bloc reconfigurable. D'autres architectures telles que Piperench [36] ou RaPiD [33] peuvent être reconfigurées à un plus haut niveau, respectivement au niveau opérateur et fonctionnel. Le concept de grain de calcul constitue en effet un sujet de recherche intéressant et important. La plupart des circuits FPGA sont « grain fin » puisqu'ils peuvent être reconfigurés au niveau du bit, ce qui contraste avec la façon dont les processeurs programmables manipulent des mots entiers (mots de 32 bits pour nombre d'entre eux). Lorsque la reconfiguration au niveau du bit n'est pas requise par l'application, des structures de grains plus gros doivent être construites à partir des blocs élémentaires de la structure reconfigurable, ce qui se traduit par un surcoût du circuit, prix de la flexibilité. Pour limiter ce surcoût, de nouvelles architectures reconfigurables de grain plus gros sont proposées. Il en résulte des structures dont les blocs élémentaires correspondent à des unités arithmétiques et logiques, des multiplieurs, des mémoires, etc. Outre Piperench et RaPiD déjà mentionnés, peuvent être par ailleurs citées les architectures : Matrix [34] au MIT, MorphoSys[44] à l'université de Californie, Irvine et parmi les réalisations commerciales, le tableau d'unités arithmétiques et logiques reconfigurables de Elixent¹², et les processeurs XPP de PACT.

Nombre de ces architectures abordent la reconfiguration d'une manière beaucoup plus efficace que ne le font les circuits FPGA, sans pour autant satisfaire le compromis performance/flexibilité/consommation précédemment évoqué. En résumé, parmi les descriptions de projets sur les architectures reconfigurables les

¹¹<http://www.chameleonsystems.com>

¹²<http://www.elixent.com/>

objectifs de « haute performance », « flexibilité » ou encore « faible consommation » sont couramment mis en avant mais ces derniers ne sont jamais réunis au sein d'une même étude. L'architecture Dart étudiée vise à satisfaire l'ensemble de ces contraintes.

4. Domaines d'application

4.1. Panorama

Le domaine d'applications privilégié est celui des télécommunications mobiles de troisième et quatrième générations.

Les autres domaines d'applications considérés sont le codage vidéo, le filtrage adaptatif, l'indexation d'images et le chiffrement (algorithme AES).

4.2. Télécommunications mobiles

Les futures générations de télécommunications constituent un domaine applicatif de premier choix pour les concepteurs de circuits intégrés en raison de la diversité des contraintes à satisfaire. Outre le très haut niveau de performance - supérieur à 12 milliards d'opérations par seconde - résultant de l'association de capacités multimédia et de techniques d'accès évoluées telles que le WCDMA que devront supporter ces systèmes (dits 3G), s'ajoute la nécessité de supporter l'ensemble des algorithmes intégrés aux normes de générations actuelles (GSM, DECT, IS-95) ainsi que leurs évolutions.

Du point de vue des architectures matérielles, les systèmes de prochaines générations auront successivement à traiter des applications très différentes. En effet, les tâches qui cohabitent sur dans chaîne de communication de troisième génération manipulent des données de tailles variables selon la « distance » séparant la tâche de l'émetteur ou du récepteur, - les tâches applicatives manipulent des données de forte granularité telles que des images alors que les tâches permettant d'accéder au support de transmission travaillent sur des données codées au niveau du bit. Du fait de l'importance du spectre des applications intégrées aux futures normes de télécommunications, les traitements à appliquer à ces données seront également très diversifiés, ce qui se traduira par des motifs de calcul eux aussi très différents. Si supporter chacune de ces contraintes est un exercice à la portée de tous, les associer rend le problème beaucoup plus délicat, d'autant que les contraintes de temps de mise sur le marché imposent la définition d'outils de développement aussi portables qu'efficaces. Lorsque par ailleurs, le résultat doit consommer très peu d'énergie - inférieur à 500mW en crête,- ce problème est insoluble si l'on se limite aux solutions architecturales actuelles.

4.3. Filtrage adaptatif

De nombreuses applications liées aux télécommunications requièrent l'utilisation de circuits spécialisés parallèles. Les algorithmes de filtrage adaptatif, en particulier, se prêtent bien à ce type de réalisation. En collaboration avec l'université de Trois Rivières au Québec, collaboration soutenue au cours du temps par divers contrats, des architectures de filtres sont étudiées et synthétisées pour être mises en œuvre sur des circuits reconfigurables ou des circuits intégrés.

Deux algorithmes de filtrage adaptatif particuliers sont étudiés : un algorithme de filtrage avec adaptation retardée par la méthode des moindres carrés (DLMS), et un algorithme de filtrage avec adaptation par réseau de neurones. Dans les deux cas, les recherches effectuées consistent à synthétiser une architecture à l'aide du logiciel MMAAlpha, et à comparer l'architecture obtenue avec une version conçue avec des outils standard. La synthèse met en évidence des limitations du logiciel MMAAlpha qui conduisent à en étendre les fonctions, faisant ainsi progresser les techniques de synthèse.

4.4. Algorithme de chiffrement AES

Les réseaux haut-débit sécurisés requièrent l'utilisation de circuits spécialisés parallèles permettant une mise en œuvre efficace des algorithmes de chiffrement. Des architectures réalisant l'algorithme de chiffrement AES sont étudiées pour être mises en œuvre sur des circuits reconfigurables.

Les travaux effectués consistent à étudier la synthèse d'architectures à partir de SystemC, langage de modélisation basé sur C++, développé par les industriels pour devenir un langage standard dans le domaine de la conception conjointe matériel/logiciel. SystemC permet aussi bien une description de haut niveau d'une application (fonctionnelle) qu'une description de bas niveau, de type transfert de registres.

5. Logiciels

5.1. Panorama

Mots clés : *bibliothèque, calcul polyèdre.*

Les recherches menées par l'action R2D2 se placent dans le contexte des outils logiciels et matériels pour la conception de systèmes sur silicium. Afin de promouvoir les techniques étudiées, de nombreux prototypes logiciels de recherche sont développés (Polylib, MMAAlpha, BSS, Armor/Calife). Parmi ceux-ci sont présentés ici trois logiciels aujourd'hui diffusés : Polylib bibliothèque *open source* de calcul sur les polyèdres, MMAAlpha pour la synthèse de haut niveau et BSS plate-forme pour la conception de circuits.

5.2. PolyLib

Participants : Sorin Olaru, Patrice Quinton [correspondant], Tanguy Risset [CompSys, Inria Rhône-Alpes].

Mots clés : *conception circuit, synthèse d'architecture, faible consommation, CAO, ASIC, parallélisme de données, parallélisation automatique, bibliothèque, calcul polyèdre.*

La bibliothèque polyédrique Polylib, développée en C, est une bibliothèque *open source* de calcul sur les polyèdres convexes. Elle a été développée initialement par Hervé Le Verge et Doran Wilde à l'Inria Rennes. Elle est aujourd'hui maintenue et développée avec le LIP (ENS Lyon) et l'ICPS de l'université de Strasbourg. La manipulation des domaines utilisés dans les équations récurrentes ou des espaces d'indices décrits par les boucles imbriquées justifie l'emploi d'une telle bibliothèque. Cette bibliothèque est actuellement utilisée (indépendamment de MMAAlpha) par plusieurs organismes de recherche (en Angleterre, États Unis, Pays-bas, ainsi qu'en France).

La bibliothèque de calcul polyédrique PolyLib a fait l'objet au cours de cette année 2002 d'une opération de développement logiciel (ODL). Cette opération a permis :

- la mise en place d'un site Web clairement identifié comme le site de PolyLib ;
- la rédaction d'une documentation utilisateur et d'un système de documentation développeur produit automatiquement ;
- la mise en place d'un site CVS pour le développement multi-sites de la bibliothèque ;
- la correction de nombreuses bogues ;
- la diffusion du logiciel.

Pour en savoir plus, se référer à <http://www.irisa.fr/polylib> ou contacter Patrice Quinton.

5.3. MMAAlpha

Participants : David Cachera, Anne-Claire Guillou, Patrice Quinton [correspondant], Tanguy Risset [CompSys, Inria Rhône-Alpes].

Mots clés : *synthèse d'architecture, CAO, ASIC, programmation fonctionnelle, parallélisme de données, parallélisation automatique.*

MMAAlpha est un logiciel qui implémente des transformations sur le langage Alpha. Le langage Alpha a été proposé par Christophe Mauras lors de sa thèse en 1989. L'implémentation est réalisée dans le langage Mathematica (d'où le nom MMAAlpha) et s'appuie sur la bibliothèque Polylib.

Les transformations de programmes Alpha sont implémentées en utilisant les possibilités de Mathematica et de la bibliothèque Polylib. Le principe d'utilisation de ces transformations est de dériver soit une architecture, soit du code séquentiel ou parallèle à partir d'une spécification algorithmique d'un traitement. Ces transformations sont semi-automatiques, c'est-à-dire que les actions à effectuer sont indiquées par l'utilisateur mais que la transformation elle-même est exécutée par MMAAlpha. Il est possible d'effectuer une dérivation automatique par défaut mais l'expérience montre que l'espace de conception est si important que cela est rarement satisfaisant.

La méthodologie de conception est héritée de la méthode de synthèse de réseaux systoliques. Ce domaine a été longuement étudié du point de vue théorique et l'environnement MMAAlpha permet de tester les différentes stratégies de synthèse existantes, d'étudier différentes possibilités de parallélisation et de générer une description architecturale d'un circuit grâce au format AlpHard (sous-ensemble du langage Alpha). La communication avec les outils de synthèse logique se fait grâce à une traduction automatique du format AlpHard vers VHDL.

Le logiciel est déposé à l'association de protection des programmes, et a été mis sous licence GNU. Il a été le support d'implémentation de nombreuses thèses réalisées à l'Irisa. Il est utilisé par quelques équipes de recherche dans le cadre de collaborations avec R2D2. Actuellement c'est un des seuls outils permettant de décrire un algorithme et son implémentation matérielle dans le même langage et de déduire cette implémentation avec des transformations sûres.

Pour en savoir plus, se référer à <http://www.irisa.fr/cosi/ALPHA/> ou contacter Patrice Quinton.

5.4. BSS

Participants : Daniel Chillet [correspondant], Sébastien Pillement, Olivier Sentieys.

Mots clés : *conception circuit, synthèse d'architecture, faible consommation, placement.*

La plate-forme logicielle de conception de circuit BSS (*Breizh Synthesis System*) propose un ensemble d'outils pour la saisie de description d'application (en VHDL ou en C), la compilation, la simulation et la synthèse d'architecture. Celle-ci est développée de façon à rendre les outils accessibles par Internet. Un mécanisme d'identification permet d'assurer, à chaque utilisateur qui accède aux outils, un espace de travail propre.

La plate-forme est actuellement composée des modules suivants.

- Un ensemble de programmes (compilateur C et VHDL, allocation, sélection, ordonnancement, génération de code) permettant la synthèse d'architectures de circuits ou processeurs spécialisés.
- Des interfaces graphiques *PUDesigner* et *GFDesigner*, permettant la visualisation et la manipulation des graphes flots de données et des architectures.
- Un outil d'estimation de la puissance au niveau architectural, *PowerCheck*, opérant à partir des architectures générées par la synthèse. Il utilise également en entrée un fichier de paramètres qui permet de caractériser la technologie du circuit et les capacités physiques des puces électroniques. Le signal peut, quant à lui, entrer en jeu de deux manières différentes : soit par ces probabilités suivant un modèle (bruit blanc, DBT), soit sous forme d'un fichier de vecteurs dont sont extraites les caractéristiques probabilistes. En sortie, *PowerCheck* fournit un rapport indiquant les puissances moyennes dissipées par chacune des parties des unités de traitement et de contrôle. *PowerCheck* donne également les puissances dissipées cycle par cycle par les différents modules.
- Un outil d'estimation de la surface et du temps des interconnexions *Jffloorplanner* opérant au niveau architectural. Le point d'entrée de l'outil est constitué d'une *netlist* générée par BSS. Cette *netlist* contient l'ensemble des informations relatives aux composants ainsi qu'aux interconnexions. L'outil fournit des indications concernant la surface finale du plan de masse, les longueurs des interconnexions ainsi que les délais d'interconnexions liés à ces longueurs. Un affichage du plan de masse estimé est disponible et peut être utilisé afin de réaliser rapidement le placement routage avec les outils de CAO standard.

Pour en savoir plus, se référer à <http://archi.enssat.fr/bss> ou contacter Daniel Chillet.

6. Résultats nouveaux

6.1. Nouvelles architectures et technologies

Participants : Michel Aline, Daniel Chillet, Raphaël David, Hélène Dubois, Ekué Kinvi-Boh, Sébastien Pillement, David Saillé, Olivier Sentieys.

Mots clés : *architecture reconfigurable, grain de calcul, architecture faible consommation, logique multivale.*

Les travaux concernent l'étude de nouvelles organisations de structures reconfigurables associant haute performance et faible consommation. Ils portent par ailleurs sur l'étude d'architectures et de circuits en logique à valeurs multiples.

6.1.1. Architecture faible consommation

Participants : Daniel Chillet, David Saillé, Olivier Sentieys.

L'apparition de la notion de mobilité durant ces dernières années a bouleversé le domaine de la conception de circuit. En effet, la contrainte de consommation liée aux systèmes embarqués est devenue prépondérante. Dans ce cadre, nous avons développé une méthodologie et un outil d'estimation de la consommation à un haut niveau d'abstraction, PowerCheck (thèse de M. Denoual [9]). Cet outil permet l'extraction d'informations sur la consommation des différentes unités mises en place dans le circuit et ceci à partir d'une description de haut niveau de l'application. Cette estimation permet alors de vérifier la satisfaction des contraintes et éventuellement d'écarter certaines solutions.

6.1.2. Architecture reconfigurable Dart

Participants : Daniel Chillet, Raphaël David, Sébastien Pillement, Olivier Sentieys.

Les travaux concernent la définition d'un modèle architectural, appelé architecture Dart, associant haute performance et faible consommation. Afin d'autoriser le partitionnement d'un système de télécommunication 3G en tâches bien distinctes (traitement d'images, de la parole, WCDMA), l'architecture se décompose en *clusters* indépendants. Chaque *cluster* dispose de son propre contrôleur et de ses propres ressources de mémorisation.

Au plus haut niveau d'abstraction de cette architecture, un contrôleur de tâches est chargé de distribuer différentes configurations aux *clusters* sous contraintes de disponibilités de ressources et d'urgence. Ces configurations se traduisent au niveau des *clusters* par la modification du programme du contrôleur et par le chargement des données manipulées dans les mémoires internes au *cluster*. Le programme de ce dernier est constitué d'une succession de configurations qui sont distribuées dynamiquement et qui peuvent concerner deux types de cible. En effet, afin que le grain des ressources de calcul soit en adéquation avec celui de la tâche implémentée, deux types de ressources cohabitent dans chacun des *clusters*. Chaque *cluster* dispose ainsi d'un cœur de FPGA pour les traitements de grain fin, tels que les codeurs de canal, et de plusieurs *DataPath Reconfigurables* (DPR) pour les traitements arithmétiques tels que les traitements de parole, d'image. Ces DPR s'organisent autour d'unités fonctionnelles (Multiplieurs, unités arithmétiques), de mémoires et de contrôleurs locaux (dédiés à la génération d'adresses) interconnectés d'une manière totalement flexible par l'intermédiaire d'un réseau multi-bus. Par ailleurs, afin d'adapter le parallélisme de l'architecture à celui de l'application, ces DPR sont interconnectés au sein des *clusters* via un réseau en grille segmenté. Ils peuvent être interconnectés afin d'implémenter un traitement critique sous une forme massivement parallèle ou être déconnectés pour travailler indépendamment sur différents *threads*. Par le biais de ce réseau segmenté, les parallélismes d'instructions et de tâches de l'architecture peuvent donc être dynamiquement adaptés aux besoins de l'application.

Afin de couvrir le très large spectre des applications intégrées aux futures normes de télécommunications, la flexibilité de l'architecture peut être adaptée au besoin de l'application en cours d'exécution. Ainsi, les applications régulières, e.g. cœurs de boucles, dans lesquelles les configurations sont utilisées pendant de longues

périodes sont distinguées des applications irrégulières où elles se succèdent rapidement, éventuellement à chaque cycle, afin d'envisager deux modes de reconfiguration.

- La reconfiguration matérielle est utilisée pour les applications régulières et assure une totale flexibilité au sein du DPR afin d'optimiser le motif d'interconnexion pour le traitement à implémenter.
- La reconfiguration logicielle cible les applications irrégulières. Afin d'autoriser une reconfiguration en 1 cycle du DPR, la flexibilité de ce dernier est alors limitée, en considérant que les unités fonctionnelles ne traitent que des opérations de type *Read-Modify-Write*, à la manière des DSP.

La synthèse d'un *cluster* a été réalisée et a abouti à une fréquence de fonctionnement de l'ordre de 130 MHz et une efficacité énergétique au pire cas de 50 MOPS/mW à raison de 3.12GIPS/*cluster*. Ces deux unités intégrant les concepts de SIMD ainsi que des décaleurs, il est possible de réaliser 4 opérations par cycle sur chaque unité pour une efficacité énergétique crête de 350 MOPS/mW à raison de 12.5 MOPS/*cluster*.

6.1.3. Architectures et circuits en logique à valeurs multiples (MVL)

Participants : Michel Aline, Daniel Chillet, Hélène Dubois, Ekué Kinvi-Boh, Sébastien Pillement, Olivier Sentieys.

L'étude consiste à concevoir l'architecture d'un processeur de traitement du signal dans une technologie ternaire et quaternaire en se basant sur une approche hiérarchique. La technologie considérée (SUS-LOC) est fondée sur une extension de la structure CMOS utilisant un agencement de transistors spécifiques au radix considéré [46]. La principale difficulté, qui est également la motivation principale de ces travaux de recherche, est engendrée par le fait que cette technique est totalement nouvelle. Il n'existe à ce jour aucune méthodologie, ni aucun outil de CAO électronique travaillant dans un autre format que le binaire. Sur l'année 2002, les travaux suivants ont été réalisés.

- Nous avons défini et validé un ensemble de cellules de base au niveau transistor (min, max, fonctions d'addition et de multiplication de base, registre, multiplexeur, mémoire, etc.).
- Ces cellules ont été caractérisées en performance temporelle, consommation et surface par des outils développés spécifiquement dans l'équipe.
- Des modèles de simulation au niveau logique utilisant des spécifications VHDL génériques ont été développés. Ces modèles intègrent les temps de propagation et la consommation électrique dans la simulation.
- Nous avons conçu et caractérisé au niveau architectural les fonctions de base d'un DSP (multiplicateur, unité arithmétique, bancs de registres, mémoire, etc.). Les résultats obtenus, en particulier sur les opérateurs arithmétiques et les mémoires, montrent que les structures ternaires offrent une meilleure efficacité énergétique que leur équivalent binaire.
- Une étude a porté sur la réalisation d'un bus d'interconnexion haute performance sur silicium utilisant une technologie quaternaire. Il a été démontré que l'énergie dissipée par ce réseau est divisée par deux.
- Une validation architecturale du fonctionnement de l'ensemble du processeur a été réalisée par simulation.
- Des outils de CAO en Java, en partie basés sur MVSIS de l'université de Berkeley, ont été développés pour automatiser la conception logique vers les schémas transistors.

6.2. Synthèse d'accélérateurs parallèles dédiés

Participants : David Cachera, Steven Derrien, Anne-Claire Guillou, Gautam Gupta, Katell Morin-Allory, Patrice Quinton, Sanjay Rajopadhye, Charles Wagner.

Mots clés : *synthèse d'architecture, CAO, ASIC, FPGA.*

Les travaux concernent le développement de méthodologies pour la conception d'accélérateurs matériels parallèles dédiés pour les parties de calcul intensif des applications et leur implantation sur des circuits programmables de type FPGA.

6.2.1. Techniques de partitionnement pour l'implantation de réseaux de processeurs sur FPGA

Participants : Steven Derrien, Patrice Quinton, Sanjay Rajopadhye.

Les outils de synthèse de réseaux de processeurs comme MMAAlpha sont utilisés pour produire, à partir d'une spécification de nid de boucles, des architectures matérielles spécialisées parallèles, dont la structure régulière est particulièrement bien adaptée à une implantation sur des circuits programmables de type FPGA. Une des principales faiblesses de ces techniques de parallélisation est l'impossibilité de contrôler précisément les caractéristiques (surface, performance, consommation) des circuits produits.

Les travaux réalisés (thèse de S. Derrien [10]) proposent des post-transformations dites « de partitionnement » qui permettent de modifier l'architecture obtenue à l'issue de la parallélisation. Cette étape de partitionnement permet notamment de choisir le nombre de processeurs formant le réseau, de contrôler précisément la réutilisation de données au sein de l'architecture, et d'adapter la structure interne des processeurs à la granularité de calcul des circuits FPGA.

Des modèles haut-niveaux qui permettent d'estimer les caractéristiques de l'architecture (surface, performance, consommation) en fonction des paramètres et du type de partitionnement utilisé ont été développés. Ces modèles ont fait l'objet d'expérimentations sur une plate-forme à base d'un FPGA Virtex de la société Xilinx, et les résultats obtenus ont permis de confirmer la validité de l'approche.

6.2.2. Génération de code VHDL

Participants : Anne-Claire Guillou, Patrice Quinton.

Un nouveau générateur de code VHDL a été développé pour répondre aux besoins d'implémentation et de test des architectures étudiées. Des modifications et des optimisations ont été apportées au générateur notamment concernant la précision des calculs en arithmétique à virgule fixe. Une autre extension majeure a été ajoutée à ce générateur de code. Il est en effet actuellement possible avec les outils de synthèse commerciaux de générer des réseaux de processeurs 2D mais cette possibilité n'était pas encore exploitée par le générateur VHDL de MMAAlpha. Une syntaxe du langage VHDL décrivant de tel réseaux a été établie et intégrée. Le back-end de l'outil MMAAlpha offre maintenant la possibilité de générer des architectures systoliques à deux dimensions.

6.2.3. Génération automatique d'interfaces

Participants : Steven Derrien, Patrice Quinton, Tanguy Risset [CompSys, Inria Rhône-Alpes], Charles Wagner.

Les travaux sont centrés sur l'utilisation de plates-formes à base de circuits FPGA. Ils ont porté sur l'interfaçage des architectures générées par les descriptions Alpha à travers le bus reliant le circuit FPGA au processeur. A partir de n'importe quel programme AlpHard représentant un réseau systolique linéaire, un traducteur génère le code VHDL de l'interface. Ces travaux ont donné lieu à une validation expérimentale sur une carte à base d'un FPGA Virtex (carte SPYDER-X2).

6.2.4. Utilisation de méthodes formelles pour la détermination de propriétés de systèmes Alpha

Participants : David Cachera, Katell Morin-Allory.

Le flot de dérivation MMAAlpha qui part d'une spécification de très haut niveau pour arriver à une description architecturale garantit théoriquement la correction de l'architecture finale. Ceci est dû au fait que les transformations utilisées conservent la sémantique des systèmes. Cependant, il est parfois nécessaire de valider certaines transformations (qui ne préservent pas intégralement la sémantique), ou de vérifier certaines propriétés non triviales de spécifications complexes. Nous avons poursuivi le travail sur les méthodes de vérification en définissant et implémentant une interface entre MMAAlpha et le système **Coq**, qui permet d'améliorer l'automatisation des preuves. En particulier, nous générons pour chaque système un principe d'induction complexe correspondant à la structure de ses dépendances. Pour développer des preuves hiérarchiques sur des systèmes structurés, nous utilisons un mécanisme d'import/export des théorèmes afin d'automatiser la réutilisation des propriétés déjà prouvées.

6.2.5. Méthodologie pour la détermination de la largeur des chemins de données

Participant : David Cachera.

La détermination de la largeur du chemin de données passe par une traduction d'un programme Alpha vers une forme abstraite (exprimée également en Alpha) qui consiste en un système d'équations sur les tailles des mots codant chaque variable. En collaboration avec Tanguy Risset (CompSys, Inria Rhône-Alpes), nous avons développé une méthode originale qui, sous certaines conditions, utilise des propriétés de l'algèbre (\max , $+$) pour dériver automatiquement une expression symbolique (dépendant des paramètres du système) pour la taille en bits de chaque variable [16].

6.2.6. Travaux sur la sérialisation

Participants : Gautam Gupta, Sanjay Rajopadhye, Patrice Quinton.

De nombreux calculs peuvent être modélisés par des systèmes d'équations récurrentes affines définis sur des domaines polyédriques. Nous étudions le problème de l'ordonnancement des calculs élémentaires de systèmes d'équations récurrentes affines en présence de réductions. Une réduction se traduit par l'utilisation d'un opérateur commutatif et associatif et par conséquent n'impose aucun ordre spécifique. Cependant, du point de vue de l'exécution sur une machine réelle, les opérateurs ont un nombre borné d'entrée, et un ordre des accumulations, encore appelé sérialisation, est nécessaire. Des sérialisations arbitraires peuvent affecter le temps d'exécution d'un programme. Nous avons proposé un algorithme qui permet de déterminer efficacement la sérialisation de toutes réductions [23].

6.3. Exploration, estimation, prototypage pour la conception de systèmes sur silicium

Participants : Imène Benkermi, François Charot, Daniel Chillet, Raphaël David, Ludovic L'Hours, Daniel Menard, Madeleine Nyamsi, Laurent Perraudeau, Sébastien Pillement, Taofik Saidi, Olivier Sentieys.

Mots clés : *synthèse d'architecture, compilation flexible, modélisation d'architecture, conception d'ASIP, arithmétique virgule fixe.*

La mise en œuvre d'une application sur une telle plate-forme reconfigurable nécessite la mise en place d'un ensemble de techniques variées (synthèse d'architecture, compilation flexible, génération de code virgule fixe, profilage, etc.) qui, par raffinements successifs, contribuent aux choix d'implémentation des différentes parties de l'application sur les composants de la plate-forme. Les actions de recherche que nous menons visent à mettre en place des méthodologies permettant la mise en œuvre des différentes parties de l'application sur les différents composants d'une plate-forme.

6.3.1. Langage de description d'architectures

Participants : François Charot, Ludovic L'Hours, Madeleine Nyamsi.

Pour la modélisation de processeurs spécialisés programmables, nous avons conçu le langage Armor. Armor permet la description de processeurs programmables au travers de leur jeu d'instructions. Une description de processeurs en Armor est une grammaire dont chaque dérivation est un comportement possible du jeu d'instructions du processeur. Armor décrit donc le comportement du jeu d'instructions, incluant sa sémantique, les informations temporelles, l'usage des ressources, ainsi que les possibilités de parallélisme au niveau instruction. Il s'agit d'une modélisation de nature comportementale du processeur. Nous étudions des extensions devant être apportées au langage Armor en vue de la modélisation de l'unité de contrôle de processeurs spécialisés programmables. L'objectif est donc d'étendre le langage pour permettre une modélisation combinant comportemental et structurel et pour faciliter la génération de modèles de processeurs exploitables, par exemple, dans un processus de synthèse de matériel.

6.3.2. Estimation de performances logicielles

Participant : François Charot.

L'environnement Calife est basé sur une bibliothèque de modules de production et d'optimisation de code. La flexibilité se situe à deux niveaux : flot de compilation et choix des modules de la bibliothèque. Un flot de compilation consiste en un agencement de modules choisis dans la bibliothèque, le flot de compilation le plus adapté à l'architecture du processeur cible pouvant ainsi être spécifié. Les modules sont paramétrés automatiquement à partir de la description du processeur cible en langage Armor. Des passes spécifiques à l'estimation de performances réalisant la sélection d'instructions et l'ordonnancement ont été développées. Elles ont permis la réalisation d'un estimateur pour le processeur OakDSPCore, un processeur de traitement de signal très largement utilisé dans des applications de télécommunication. Cet estimateur a été expérimenté sur des noyaux d'applications de traitement de signal typiques. Il montre des performances très proches de celles du code assembleur développé manuellement (thèse de F. Djieya [11]).

6.3.3. *Méthodologie d'implantation d'algorithmes spécifiés en virgule flottante dans des architectures virgule fixe*

Participants : François Charot, Daniel Chillet, Daniel Menard, Taoufik Saidi, Olivier Sentieys.

Dans le cadre des processeurs spécialisés, comme par exemple les processeurs de traitement de signal DSP, nous développons une nouvelle méthodologie de compilation d'algorithmes spécifiés en virgule flottante au sein d'architectures programmables en virgule fixe sous contrainte de précision de calcul [27][28]. Celle-ci vise à déterminer le codage optimal, permettant de maximiser la précision, et de minimiser le temps d'exécution et la taille du code.

Dans un premier temps, la dynamique des données est déterminée à l'aide d'une approche analytique combinant deux techniques différentes. Ces informations sur la dynamique permettent de déterminer la position de la virgule de chaque donnée en tenant compte de la présence éventuelle de bits de garde au sein de l'architecture. Pour obtenir un format des données en virgule fixe complet, la largeur de chaque donnée est déterminée en prenant en compte l'ensemble des types des données manipulées au sein du DSP. La méthode sélectionne la séquence d'instructions permettant de fournir une précision suffisante en sortie de l'algorithme et de minimiser le temps d'exécution du code. La dernière phase du processus de codage correspond à l'optimisation du format des données en vue d'obtenir une implantation plus efficace. Les différentes opérations de recadrage sont déplacées afin de minimiser le temps d'exécution global tant que la précision en sortie de l'algorithme est supérieure à la contrainte. Deux méthodes ont été mises en œuvre en fonction des capacités de parallélisme au niveau instruction de l'architecture ciblée.

La précision de la spécification en virgule fixe est évaluée à travers la détermination du rapport signal à bruit de quantification (RSBQ). Une méthodologie permettant de déterminer automatiquement l'expression analytique du RSBQ en fonction du format des données en virgule fixe a été définie (thèse de D. Menard [12]). Cette méthodologie permet de traiter les systèmes linéaires [28] et les systèmes non-linéaires et non-récurrents.

La structure de compilation utilisée pour mettre en œuvre cette méthodologie est composée d'une partie frontale Suif et du générateur de code Calife. L'outil implantant la méthodologie est en cours de développement. Une première version permet d'obtenir une spécification en virgule fixe optimisée dans le cadre des processeurs ne possédant pas de parallélisme au niveau instruction.

6.3.4. *Environnement logiciel de l'architecture Dart*

Participants : Imène Benkermi, François Charot, Daniel Chillet, Raphaël David, Sébastien Pillement, Taoufik Saidi, Olivier Sentieys.

Outre le respect des contraintes de faible consommation, l'aspect faisabilité d'une chaîne de compilation a également été pris en compte dans la conception de l'architecture Dart. Un modèle de simulation en SystemC ainsi qu'un modèle VHDL synthétisable ont été réalisés et une chaîne de compilation est en cours de développement.

A partir d'un graphe flot de données et de contrôle (CDFG) décrivant l'application, un partitionnement permet de distinguer les parties du graphe vues comme des reconfigurations matérielles de celles qui sont compilées comme une suite de configurations logicielles. Dans tous les cas, il est nécessaire de générer le

code des générateurs d'adresses afin d'alimenter les DPR en données au travers de leurs mémoires locales. La stratégie utilisée actuellement consiste à transformer le CDFG puis à synthétiser le cœur de boucle vers une configuration matérielle. Ces travaux reposent sur des concepts issus de la synthèse d'architecture ou du systolique.

L'outil de compilation est basé sur l'utilisation conjointe de Suif, du compilateur recible Calife, et de l'outil de synthèse de haut niveau BSS. Le graphe CDFG est généré à partir de la représentation intermédiaire de Suif. Des passes spécifiques de déroulage de boucles et d'extraction des cœurs de boucles ont été développées. Les outils de génération des configurations logicielles (cDART) et d'adresses (ACG) exploitent des passes classiques de compilation qui sont réunies au sein de l'environnement Calife. Les configurations matérielles (gDART) sont quant à elles générées en s'appuyant sur l'infrastructure BSS afin d'obtenir un DFG du cœur que gDART optimise et assigne sur les différents opérateurs de l'architecture. En sortie de ces différents outils, un simulateur développé en SystemC permet de valider l'implémentation réalisée et d'évaluer son efficacité énergétique.

6.4. Etude d'applications

Participants : Alexandre Buisson, Raphaël David, Michel Guitton, Daniel Menard, Sébastien Pillement, Patrice Quinton, Pascal Scalart, Olivier Sentieys.

Mots clés : *télécommunication mobile, WCDMA, codage vidéo, indexation d'image.*

6.4.1. Prototypage d'applications de télécommunications mobiles

Participants : Raphaël David, Michel Guitton, Daniel Menard, Sébastien Pillement, Pascal Scalart, Olivier Sentieys.

Une chaîne complète de communication mobile de troisième génération, utilisant des codeurs sources (vidéo, audio, parole) et des techniques d'accès et de modulation (WCDMA), a été réalisée dans le cadre de la collaboration avec STMicroelectronics (spécifications SPW, Matlab et C). Une spécification mixte C/SystemC a également été réalisée. Ces spécifications système permettent d'obtenir des métriques de performances lors de la simulation SystemC (opérations, accès mémoire, consommation) pour mettre en avant l'intérêt des architectures et des méthodes développées. La complexité de ces algorithmes - pouvant atteindre plusieurs dizaines de GOPS - nécessite des études d'architectures spécifiques à haute performance et faible énergie dissipée. Plusieurs cibles architecturales ont été considérées (processeurs DSP VLIW Lx de STMicroelectronics et C64x de Texas Instruments, architecture Dart) et des évaluations de performance ont été effectuées.

Une détection multi-utilisateurs WCDMA a par ailleurs été décrite sous Matlab.

La conception architecturale d'une chaîne de communication E/R WCDMA sur FPGA Virtex a été étudiée puis validée sur une carte Virtex (RC1000 de Celoxica).

6.4.2. Codage d'images et de vidéo

Participants : Alexandre Buisson, Olivier Sentieys.

Une estimation des performances de JPEG2000 à partir d'un code C++ a été réalisée. Certaines parties (filtrage, transformée en ondelettes) ont été implémentées sur le processeur Lx de STMicroelectronics.

Les travaux ont par ailleurs porté sur l'intégration de techniques de codage vidéo par maillages dans des schémas de codage robuste et la mise en adéquation de ces algorithmes avec une architecture multi-processeurs Pentium. La complexité des algorithmes basés maillage est importante et leur implémentation efficace représente un point critique. Nous avons donc exploré les techniques à mettre en œuvre pour améliorer les performances de ceux-ci, afin d'obtenir des performances temps-réel permettant de les hybrider efficacement avec des codeurs en voie de standardisation (MPEG4 et H264) et dans le but de les faire émerger dans un contexte industriel (thèse de A. Buisson [8]).

6.4.3. Indexation d'images

Participant : Patrice Quinton.

Les algorithmes d'indexation d'images consistent à retrouver une image sur la base de son contenu. Les méthodes se basent sur l'élaboration d'un descripteur qui caractérise l'image suivant plusieurs critères. Les recherches portent alors sur les descripteurs, et non sur l'image brute. Rechercher les images « similaires » à une image donnée consiste à comparer tous les descripteurs de la base au descripteur de cette image. Un descripteur d'image est un ensemble de vecteurs. Il faut donc calculer une distance entre des ensembles de vecteurs pour juger de la proximité de deux images. C'est un calcul simple, mais répétitif qui justifie l'utilisation d'architectures régulières qui peuvent être synthétisées à l'aide du logiciel MMAAlpha. Les travaux sur ce thème font l'objet de collaboration avec Symbiose et Texmex. Ils portent sur l'étude d'accélérateurs matériels dédiés conçus avec MMAAlpha - thèse en co-tutelle avec l'université du Yaoundé (Cameroun) de Auguste Noumsi.

7. Contrats industriels

7.1. Hades : PEA sécurisation des réseaux hauts débits (2002-2003)

Participants : François Charot, Ludovic L'Hours, Charles Wagner.

Les travaux qui sont réalisés par l'équipe concernent une étude de mise en œuvre des algorithmes de sécurité (algorithmes de filtrage et cryptographie) et leur adaptation au contexte du haut débit, avec une focalisation plus particulière sur l'intégration et l'optimisation des algorithmes de chiffrement : respect des exigences du haut débit (gigabit/s) de l'algorithme de chiffrement AES [45], étude prospective sur le téra-bit/s. Le second aspect abordé concernera les spécifications de l'architecture d'un composant chiffre répondant aux contraintes du haut débit. Ces travaux s'appuient sur les recherches menées avec les outils MMAAlpha pour la parallélisation de l'algorithme AES, ainsi que sur Calife/Armor pour la définition du composant chiffre.

7.2. IST Ozone (2002-2004)

Participants : François Charot, Madeleine Nyamsi, Patrice Quinton, Charles Wagner.

Le projet IST Ozone (*New Technologies and services for emerging nomadic societies*) a débuté en novembre 2001. Il regroupe les partenaires suivants : Philips Electronics Nederland (Pays-Bas), Imec (Belgique), Epictoid (Pays-Bas), Philips Research Laboratory (France), Eindhoven University of Technology (Pays-Bas), Inria, Thomson Multimedia (France).

Le projet Ozone a pour objectif d'investiguer, de définir, d'implémenter et d'intégrer une plate-forme générique pour des applications d'intelligence ambiante. Ce projet a pour objectif de rendre plus conviviales les interactions de l'utilisateur avec les appareils et les applications pour permettre de nouveaux services de meilleure qualité. L'un des axes de recherches concerne les architectures matérielles sur lesquelles peut s'appuyer l'intelligence ambiante.

L'objet des travaux menés par le projet concerne l'intégration des outils MMAAlpha et Calife, et son utilisation pour la compilation de parties coûteuses en calculs d'applications.

7.3. PHRASE : reconfigurabilité et processeurs VLIW dans les architectures parallèles hétérogènes

Participants : Daniel Chillet, Raphaël David, Sébastien Pillement, Olivier Sentieys.

PHRASE est supporté par des contrats du Ministère de l'industrie et des finances (MEFI/STSI) depuis 1999. Les travaux sont liés à des études amont visant des circuits intégrés de nouvelle génération complètement programmables et reconfigurables. Les études sont menées dans le cadre d'une coopération entre STMicroelectronics, et l'équipe AS de l'université de Bretagne Occidentale.

7.4. Architectures basée sur la logique à valeurs multiples pour applications en télécommunications (2001-2004)

Participants : Michel Aline, Daniel Chillet, Hélène Dubois, Ekué Kinvi-Boh, Sébastien Pillement, Olivier Sentieys.

Jusqu'au développement de la technologie SUS-LOC par E.D. Olson, les techniques MVL n'étaient pas réalisables en pratique et restaient seulement théoriques. Cette technologie permet de réaliser toute fonction MVL avec une complexité équivalente à la technologie CMOS dans le cas binaire et utilise les fonderies de circuits standard. Le brevet international la protégeant a été classé comme un des plus innovants de ces dernières années par le US Patent Office.

Une collaboration de recherche active a été établie avec EDO LLC, la société américaine fondée par D. Olson, pour l'étude et le développement de nouveaux systèmes utilisant la MVL, en particulier dans le domaine des processeurs de traitement du signal (DSP) pour les télécommunications.

8. Actions régionales, nationales et internationales

8.1. Actions nationales

F. Charot est membre du comité de pilotage du Réseau Thématique Pluridisciplinaire « Systèmes Complexes Intégrés sur Puces (SOC) », mis en place au département STIC du CNRS. Il est co-animateur de l'une des actions spécifiques de ce programme thématique, l'action « Architectures reconfigurables dynamiquement ».

Les membres de l'équipe participent aux actions spécifiques du Réseau Thématique Pluridisciplinaire SOC : « Architectures reconfigurables dynamiquement » (AS27), « Spécification et création d'une bibliothèque de blocs IP » (AS28), « Systèmes d'exploitation et architectures multiprocesseurs » (AS29), « Méthodes formelles de vérification » (AS55), ainsi qu'aux autres actions spécifiques suivantes : « radio-logicielle » (AS37), « arithmétique des ordinateurs » (AS78) et « compilation pour systèmes embarqués » (AS82).

Les membres de l'équipe participent aux activités :

- du GdR-PRC ISIS (Information Signal ImageS), groupe de travail GT7 « Adéquation Algorithmes Architectures ». Participation active à l'opération 7.2 (VLSI et codage) du GdR ;
- du GdR CAO (Conception Assistée par Ordinateur) ;
- du GdR-PRC ARP (Architectures Réseaux et Parallélisme), groupe de travail « Architectures Spécialisées » ;
- de l'action incitative inter GdR ISIS et GdR ARP « Systèmes Reconfigurables ».

8.2. Relations bilatérales internationales

8.2.1. Europe

R2D2 coopère avec l'Université de Leiden aux Pays-Bas (Ed Deprettere) sur les techniques de synthèse d'architectures parallèles,

R2D2 coopère avec l'UCL de Louvain sur le thème des circuits intégrés en technologie ternaire. Un circuit prototype est en cours de développement sur la technologie SOI du laboratoire de micro-électronique (DICE) de l'UCL.

8.2.2. Afrique

Dans le cadre du projet Cari et du programme FICU de l'Agence Universitaire pour la Francophonie, R2D2 coopère avec l'université de Yaoundé (Cameroun) sur l'algorithmique systolique.

R2D2 coopère avec l'ENIT de Tunis sur le thème des architectures pour les télécommunications mobiles.

R2D2 coopère avec l'université de Tananarive à Madagascar, pour la formation à la recherche d'enseignants-chercheurs.

8.2.3. Amérique du Nord

R2D2 coopère avec le laboratoire LSSI de l'Université du Québec à Trois-Rivières, sur la conception d'architectures pour des filtres.

R2D2 entretient des relations avec le département d'informatique de l'Université du Colorado à Fort-Collins sur le développement de MMAAlpha.

R2D2 coopère avec le laboratoire LRTS de l'université de Laval à Québec sur le thème des architectures pour la radio-logicielle.

9. Diffusion des résultats

9.1. Animation de la communauté scientifique

P. Quinton a été membre du comité exécutif de DATE 2002, et du comité de programme de ASAP 2002.

O. Sentieys a été membre du comité de programme de SympA (Symposium en Architectures de Machines) et de la conférence ISQED (IEEE International Symposium on Quality of Electronic Design).

Sébastien Pillement a été membre du comité de programme de la *12th International Conference on Field Programmable Logic and Application FPL*, 2002.

S. Pillement, P. Quinton, et O. Sentieys ont été membres du comité d'organisation et de programme du colloque CAO de circuits et systèmes, 2002.

O. Sentieys est rapporteur pour les revues internationales IEEE Trans. on Signal Processing, et Journal of VLSI Signal Processing.

P. Scalart est rapporteur pour les revues internationales IEEE Trans. on Signal Processing, IEEE Trans. on Speech & Audio Processing, IEEE Signal Processing Letters, Speech Communication.

O. Sentieys est rapporteur pour la revue française Technique et Science Informatiques (TSI)

F. Charot et O. Sentieys sont rapporteurs pour la revue Traitement du Signal.

P. Quinton a été invité au Workshop System Architecture MOdelling and Simulation, en juillet 2002 à Samos, Grèce. Il a été orateur invité à la conférence ASCII 2002.

9.2. Enseignement universitaire et responsabilités

F. Charot est responsable d'un cours sur les « Applications de l'architecture dans les télécommunications » en DIIC.

D. Chillet enseigne un cours sur les « Architectures des processeurs évolués » au DEA STIR.

H. Dubois est directrice des études adjointe de l'Enssat.

M. Guitton est responsable de la communication à l'Enssat.

L. Perraudeau est directeur adjoint de l'Ifsic (responsable du budget et des équipements informatiques). Il est responsable d'un cours sur les langages objet dans le DESS Isa (Informatique et ses applications) de l'université de Rennes 1, enseigne la conception de circuits intégrés en DIIC deuxième année), et intervient en Licence d'informatique, en Deug Sciences, mention SM et STPI ainsi qu'en formation continue.

P. Quinton est directeur de l'Ifsic. Il est responsable du cours algorithmique parallèle (module Alpha) dans le DEA d'informatique de l'université de Rennes 1, enseigne en Deug Sciences, mention SM et STPI, ainsi qu'en DIIC (seconde et troisième année).

P. Scalart enseigne un cours sur les télécommunications mobiles à l'École Nationale des Ingénieurs de Tunis - ENIT.

O. Sentieys responsable de la filière signal (orientation 2) du DEA STIR de l'Université de Rennes 1 et du DRT électronique de l'Enssat. Il enseigne un cours sur les « Méthodologies de conception des systèmes intégrés » au DEA STIR et sur la « Réduction de consommation des circuits CMOS numériques » en 3ème année Enst de Bretagne.

9.3. Accueil d'étudiants stagiaires

L'action R2D2 a accueilli les stagiaires suivants : Deepanjyoti Sarkar (IIT Kharagpur, Inde), Auguste Noumsi Woguia (université de Yaoundé, Cameroun), Joelson Solofonia (université de Tananarive, Madagascar), Eslam Yahya (ITI, Egypte), Ludovic L'Hours, David Pichardie, Anne-Sophie Adde (Ifsic), Ahmed Amara (université de Bretagne Sud), Wael Bazarbachi, Mostafa Maksoud (Faculté de Génie Électrique du Liban) Triki Sami, Fatten Ben Abdalah (ENI de Tunis), Barbara Garcia (université technique de Zaragoza).

10. Bibliographie

Bibliographie de référence

- [1] F. CHAROT, G. LE FOL, P. LEMONNIER, C. WAGNER, C. BOUVILLE, R. BARZIC. *Towards Hardware Building Blocks for Software-Only Real Time Video Processing : the MOVIE Approach*. in « IEEE Transactions on Circuits and Systems for Video Technology », numéro 6, volume 9, September, 1999.
- [2] C. MAURAS. *Alpha : un langage équationnel pour la conception et la programmation d'architectures parallèles synchrones*. Thèse, Université de Rennes 1, décembre, 1989.
- [3] V. MESSÉ. *Production de compilateurs flexibles pour la conception de processeurs programmables spécialisés*. thèse, Université de Rennes 1, mars, 1999.
- [4] P. QUINTON, V. V. DONGEN.. *The mapping of linear recurrence equations on regular arrays*. in « Journal of VLSI Signal Processing », volume 1, 1989, pages 93-113.
- [5] P. QUINTON, Y. ROBERT. *Systolic Algorithms and Architectures*. Prentice Hall and Masson, 1989.
- [6] S. V. RAJOPADHYE, S. PURUSHOTHAMAN, R. M. FUJIMOTO. *On Synthesizing Systolic Arrays from Recurrence Equations with Linear Dependencies*. in « Proceedings, Sixth Conference on Foundations of Software Technology and Theoretical Computer Science », Springer Verlag, LNCS 241, pages 488-503, New Delhi, India, décembre, 1986.
- [7] F. DUPONT DE DINECHIN. *Systèmes structurés d'équations récurrentes : mise en œuvre dans le langage Alpha et applications*. Thèse de doctorat, université de Rennes I, janvier, 1997.

Thèses et habilitations à diriger des recherche

- [8] A. BUISSON. *Implémentation efficace d'un codeur vidéo hiérarchique granulaire sur une architecture à processeurs multimédia*. thèse, Université de Rennes 1, octobre, 2002.
- [9] M. DENOUAL. *Estimation de haut niveau de la consommation de systèmes sur silicium*. thèse, Université de Rennes 1, octobre, 2001.
- [10] S. DERRIEN. *Techniques de partitionnement pour l'implantation de réseaux de processeurs sur FPGA*. thèse, Université de Rennes 1, décembre, 2002.
- [11] F. DJIEYA. *Contributions à la conception de systèmes de contrôle de trafic et de gestion de ressources en ATM*. thèse, Université de Rennes 1, décembre, 2002.

- [12] D. MÉNARD. *Méthodologie de compilation d'algorithmes de traitement du signal en précision infinie pour les processeurs en virgule fixe*. thèse, Université de Rennes 1, décembre, 2002.
- [13] R. YU. *Estimation de haut niveau du placement et des interconnexions dans les circuits VLSI submicroniques*. thèse, Université de Rennes 1, mai, 2002.

Articles et chapitres de livre

- [14] R. DAVID, D. CHILLET, S. PILLEMENT, O. SENTIEYS. *SOC Design Methodologies*. Kluwer Academic Publishers, 2002, chapitre A Dynamically Reconfigurable Architecture for Low-Power Multimedia Terminals, pages 51-62.
- [15] P. QUINTON, T. RISSET. *Structured Scheduling of Recurrence Equations : Theory and Practice*. in « Lecture Notes in Computer Science », volume 2268, 2002.

Communications à des congrès, colloques, etc.

- [16] D. CACHERA, T. RISSET. *Advances in Bit Width Selection Methodology*. in « Proc. Application-Specific Systems, Architectures and Processors », série IEEE, pages 381-390, San Jose, CA, juillet, 2002.
- [17] R. DAVID, D. CHILLET, S. PILLEMENT, O. SENTIEYS. *Mapping Future Generation Mobile Telecommunication Applications on a Dynamically Reconfigurable Architecture*. in « IEEE International Conference on Acoustic Speech, and Signal Processing ICASSP 2002 », mai, 2002.
- [18] R. DAVID, D. CHILLET, S. PILLEMENT, O. SENTIEYS. *A Compilation Framework for a Dynamically Reconfigurable Architecture*. in « 12th IEEE International Conference on Field Programmable Logic and Applications, FPL 2002 », volume 2438, Lecture Notes in Computer Science, Springer CS Press, septembre, 2002.
- [19] R. DAVID, D. CHILLET, S. PILLEMENT, O. SENTIEYS. *A High-Performance dynamically reconfigurable embedded architecture*. in « Sophia Antipolis Conference on Microelectronics SAME'2002 », 2002.
- [20] R. DAVID, D. CHILLET, S. PILLEMENT, O. SENTIEYS. *DART : A Dynamically Reconfigurable Architecture dealing with Next Generation Telecommunications Constraints*. in « 9th IEEE Reconfigurable Architecture Workshop RAW », IEEE CS Press, avril, 2002.
- [21] S. DERRIEN, A. GUILLOU, P. QUINTON, T. RISSET, C. WAGNER. *Automatic Synthesis of Efficient Interfaces for Compiled Regular Architectures*. in « Proc. of the System Architecture Modeling and Simulation Workshop », série Lecture Notes in Computer Science, Springer Verlag, Samos, Greece, 2002.
- [22] S. DERRIEN, S. RAJOPADHYE. *Energy/power estimation of regular processor arrays*. in « Proceedings of the 15th international symposium on System Synthesis », ACM Press, pages 50-55, 2002.
- [23] G. GUPTA, S. RAJOPADHYE, P. QUINTON. *Scheduling Reductions on Realistic Machines*. in « Proceedings of the fourteenth annual ACM Symposium on Parallel Algorithms and Architectures », ACM Press, pages 117-126, 2002.

- [24] D. MENARD, D. CHILLET, F. CHAROT, O. SENTIEYS. *Automatic Floating-point to Fixed-point Conversion for DSP Code Generation*. in « International Conference on Compilers, Architectures and Synthesis for Embedded Systems 2002 (CASES 2002) », Grenoble, octobre, 2002.
- [25] D. MENARD, P. QUEMERAIS, O. SENTIEYS. *Influence of fixed-point DSP architecture on computation accuracy*. in « XI European Signal Processing Conference (EUSIPCO 2002) », Toulouse, septembre, 2002.
- [26] D. MENARD, O. SENTIEYS. *Implantation d'algorithmes spécifiés en virgule flottante dans les DSP virgule fixe*. in « Symposium en Architectures Nouvelles de Machines SYMPA'8 », Avril, 2002.
- [27] D. MENARD, O. SENTIEYS. *A methodology for evaluating the precision of fixed-point systems*. in « IEEE International Conference on Acoustic Speech, and Signal Processing ICASSP 2002 », Orlando, mai, 2002.
- [28] D. MENARD, O. SENTIEYS. *Automatic Evaluation of the Accuracy of Fixed-point Algorithms*. in « IEEE/ACM Design, Automation and Test in Europe (DATE-02) », Paris, mars, 2002.
- [29] S. PILLEMENT, D. CHILLET, O. SENTIEYS. *A Virtual Component for Motion Estimation Algorithm*. in « ERSAS'02 : 2002 International Conference on Engineering of Reconfigurable Systems and Algorithms », juin, 2002.
- [30] P. QUINTON. *Compiling Algorithms to High-Performance Hardware : A Step towards SoC Design*. Annual Conference of the Advances School for Computing and Imaging, ASCII 2002, Lochem, The Netherlands, juin, 2002.
- [31] O. SENTIEYS, S. PILLEMENT, D. CHILLET. *Behavioral IP Specification and Integration Framework for High-Level Design Reuse*. in « ISQED 2002, IEEE International Symposium on Quality Electronic Design », mars, 2002.

Rapports de recherche et publications internes

- [32] D. CACHERA, T. RISSET. *Advances in Bit Width Selection Methodology*. rapport technique, numéro 1458, Irisa, Campus de Beaulieu, Rennes, avril, 2002.

Bibliographie générale

- [33] D. C. CRONQUIST, P. FRANKLIN, C. FISHER, M. FIGUEROA, C. EBELING. *Architecture Design of Reconfigurable Pipelined Datapath*. in « Advance Research in VLSI », 1999.
- [34] A. DEHON. *Reconfigurable Architecture for General-Purpose Computing*. thèse de doctorat, MIT, 1996.
- [35] G. EPSTEIN. *Multiple-Valued Logic Design : An introduction*. Institute of Physics Publishing, Bristol, 1993.
- [36] S. C. GOLDSTEIN, H. SCHMIT, M. BUDIU, S. CADAMBI, M. MOE, R. R. TAYLOR. *PipeRench : A Reconfigurable Architecture and Compiler*. in « IEEE Computer », avril, 2000.
- [37] T. GRÖTKER, E. MULTHAUP, O. MAUSS. *Evaluation of HW/SW Tradeoffs Using Behavioral Synthesis*. in « ICSPAT'96 », Boston, octobre, 1996.

- [38] R. HARTENSTEIN. *A Decade of Reconfigurable Computing : A Visionary retrospective*. in « Design Automation and Test in Europe (DATE) », 2001.
- [39] J. HAUSER, J. WAWRZYNEK. *GARP : A MIPS processor with a reconfigurable coprocessor*. in « IEEE Symposium on FPGAs for Custom Computing Machines », juin, 1997.
- [40] H. KEDING, M. COORS, O. LUTHJE, H. MEYR. *Fast Bit True Simulation*. in « Design Automation Conference 2001 (DAC 2001) », Las Vegas, juin, 2001.
- [41] K. KEUTZER, S. MALIK, R. NEWTON, J. RABAEY, A. SANGIOVANNI-VINCENTELLI. *System Level Design : Orthogonalization of Concerns and Platform-based Design*. in « IEEE Transactions on Computer-Aided of Circuits and Systems », numéro 12, volume 19, décembre, 2000.
- [42] K. KUM, J. KANG, W. SUNG. *AUTOSCALER for C : An optimizing floating-point to integer C program converter for fixed-point digital signal processors*. in « IEEE Transactions on Circuits and Systems II », volume 47, septembre, 2000, pages 840-848.
- [43] R. LEUPERS. *Retargetable Code Generation for Digital Signal Processors*. Kluwer Academic Publishers, 1997.
- [44] G. LU, H. SINGH, M. LEE, N. BAGHERZADEH, F. KURDAHI, E. FILHO. *The MorpoSys Parallel Reconfigurable System*. in « Euro-Par'99, LNCS 1685 », 1999.
- [45] J. NECHVATAL, E. BARKER, L. BASSHAM, W. BURR, M. DWORKIN, J. FOTI, E. ROBACK. *Report on the developement of the Advanced Encryption Standard (AES)*. rapport technique, National Institute of Standard and Technology, octobre, 2000.
- [46] E. OLSON. *Supplementary Symmetrical Logic Circuit Structure*. in « IEEE International Symposium on Multiple-Valued Logic », 1999.
- [47] S. PEES, A. HOFFMANN, V. ZIVOJNOVIC, H. MEYR. *LISA - Machine Description Language for Cycle-Accurate Models of Programmable DSP Architectures*. in « DAC 1999 », juin, 1999.
- [48] J. RABAEY. *A low-energy heterogeneous reconfigurable DSP IC*. in « Design Automation Conference (DAC) », juin, 2000.
- [49] C. RUPP, M. LANDGUTH, T. GRAVERICK, E. GOMERSALL, H. HOLT. *The NAPA Adaptative Processing Architecture*. in « IEEE Symposium on FPGAs for Custom Computing Machines », avril, 1998.
- [50] M. H. S. HAUCK, J. KAO. *The Chimera Reconfigurable Functional Unit*. in « IEEE Symposium on FPGAs for Custom Computing Machines », 1997.
- [51] A. SANGIOVANNI-VINCENTELLI, G. MARTIN. *Platform-Based Design and Software Design Methodology for Embedded Systems*. in « IEEE Design and Test of Computers », novembre, 2001.
- [52] R. SCHREIBER, S. ADITYA, S. MAHLE, V. KATHAIL, B. RAU, D. CRONQUIST, M. SIVARAMAN. *PICO-*

NPA : High-Level Synthesis of Nonprogrammable Hardware Accelerators. rapport technique, numéro HPL-2001-249, HP Laboratories Palo Alto, octobre, 2001.

- [53] M. WILLEMS, V. BURSGENS, H. KEDING, H. MEYR. *System Level Fixed-Point Design Based On An Interpolative Approach.* in « Design Automation Conference (DAC-97) », 1997.