

Proposition de post-doc

Optimisations de code au niveau source pour la synthèse d'accélérateurs matériels : transfert des données et allocation mémoire

Mots-clés : Compilation, synthèse de haut niveau, FPGA, recouvrement calculs/communications, transformations au niveau source, allocation mémoire.

Encadrants : Christophe Alias (CR2 Inria) et Alain Darté (DR2 CNRS), équipe Compsys (et en relation avec l'équipe de synthèse de STMicroelectronics, Crolles, et l'équipe Cairn, Rennes).

Laboratoire : Laboratoire de l'Informatique du Parallélisme (LIP), UMR CNRS, ENS-Lyon, UCB-Lyon, INRIA 5668, École normale supérieure de Lyon, 46, Allée d'Italie, 69007 Lyon.

Mél : Christophe.Alias@ens-lyon.fr, Alain.Darte@ens-lyon.fr

Téléphone : +33 (0)4 72 72 82 59 et +33 (0)4 72 72 85 48

Télécopie : +33 (0)4 72 72 80 80

Web : <http://www.ens-lyon.fr/LIP/COMPSYS/>

Sujet de recherche

Les équipes de recherche Compsys (Lyon) et Cairn (Rennes), en association avec STMicroelectronics, cherchent à adapter et étendre les transformations de boucles de haut-niveau (c'est-à-dire au niveau source), développées autrefois pour la parallélisation de boucles, au contexte de la synthèse de haut niveau. Le projet S2S4HLS (source-to-source for HLS) a pour but de développer une boîte à outils de transformations au niveau source pour la HLS. Ce sujet de post-doc est financé dans ce cadre.

Les applications de type multimédia, portant sur des boucles impliquant des données multidimensionnelles de grande taille, sont des candidats naturels à une mise en œuvre matérielle, sous la forme d'un composant/co-processeur dédié. Celui-ci peut être soit conçu à la main, soit obtenu directement à partir d'une spécification algorithmique, en utilisant un outil de synthèse de haut niveau (Altera/C2H, Catapult-C, Pico-Express, Ugh, ou tout autre outil). Même s'ils n'égalent pas les performances obtenues par une conception manuelle du circuit, les outils de ce type permettent de réduire considérablement les temps de conception. Pour les améliorer, deux solutions sont possibles : développer un nouvel outil de synthèse complet, intégrant de nouveaux paradigmes et de nouvelles optimisations, ou partir d'un outil existant et agir sur ses performances par des transformations en amont. C'est la seconde solution que nous choisissons car il est temps aujourd'hui d'accepter les outils de synthèse existants tels qu'ils sont, comme on le fait pour les compilateurs de programme, même si un expert pourra toujours faire mieux à la main ou avec des outils de plus bas niveau. L'objectif est donc plutôt de se concentrer sur les développements de type front-end, méthodologies, langages de spécification, pour représenter, analyser et transformer les applications en amont des outils de synthèse.

À l'heure actuelle, bien que quelques uns soient capables de gérer plus ou moins efficacement les boucles imbriquées, ces outils souffrent de certaines limitations. De plus, les utilisateurs, même les plus experts, n'ont qu'une compréhension limitée de ce que le back-end (l'outil HLS) fait et pourquoi il ne produit pas

toujours les résultats escomptés. Un effort doit être fait pour analyser les flots de conception de ces outils et mieux les exploiter. Pour cela, à partir d'une autre spécification séquentielle en C, au delà de transformations purement source à source, c'est-à-dire générant une autre spécification séquentielle en C, il semble nécessaire de générer une spécification de processus communicants, adaptée à l'outil de HLS cible. Les premiers travaux de l'équipe Compsys sur le sujet [1], dans le cadre de la thèse d'Alexandru Plesco, ont permis de montrer l'intérêt de cette approche pour l'optimisation des transferts de données depuis une mémoire DDR vers un accélérateur matériel généré automatiquement par l'outil C2H d'Altera. L'objectif de ce post-doc est d'aller au delà de ces premiers résultats, sur plusieurs points.

La méthode développée pour l'instant consiste en la génération de plusieurs accélérateurs communiquant par des mémoires locales partagées, synchronisés par des FIFOs de contrôle, et permettant le transfert de données par paquets, en recouvrant calculs et communications. La réutilisation de la mémoire locale est faite, après une phase de tiling (calcul par blocs), par une technique de repliement [2], implantée dans les outils BEE (partie analyse) et CL@K (partie optimisation) décrits dans [3]. Les analyses et transformations nécessaires pour la génération des accélérateurs multiples sont, elles, implantées dans le logiciel CHUBA [4]. Ce dernier est un premier prototype qui nécessite de nombreuses améliorations et extensions. Ces recherches devront se faire en développant également un ensemble de directives de compilation pertinentes qui enrichiront la représentation de la spécification initiale.

Plus de détails sur le travail de recherche attendu sont disponibles, sur demande, dans une version plus longue de ce document.

Expertise requise

La réalisation complète du travail nécessite des compétences diverses :

- en mathématiques et algorithmique pour la formalisation des optimisations, que ce soit pour les fonctions d'accès à la mémoire ou pour les transformations de programmes ;
- en programmation, principalement C et C++, pour le développement dans CL@K, CHUBA et leurs extensions.
- en outils pour la synthèse de circuits : outils de synthèse (CatapultC et C2H Altera notamment), simulateurs, environnements de développements pour FPGA.

Références

- [1] C. Alias, A. Darté et A. Plesco. « Optimizing DDR-SDRAM Communications at C-Level for Automatically-Generated Hardware Accelerators. An Experience with the Altera C2H HLS Tool ». Dans *21st IEEE International Conference on Application-specific Systems, Architectures and Processors (ASAP'07)* (Rennes, France, juillet 2010), IEEE Computer Society.
- [2] A. Darté, R. Schreiber et G. Villard. « Lattice-Based Memory Allocation ». *IEEE Transactions on Computers* **54**, numéro 10 (octobre 2005), 1242–1257. Special Issue : Tribute to B. Ramakrishna (Bob) Rau.
- [3] C. Alias, F. Baray et A. Darté. « Bee+Cl@k : An Implementation of Lattice-Based Array Contraction in the Source-to-Source Translator Rose ». Dans *ACM SIGPLAN/SIGBED Conference on Languages, Compilers, and Tools for Embedded Systems (LCTES'07)* (San Diego, USA, juin 2007).
- [4] A. Plesco. *Program Transformations and Memory Architecture Optimizations for High-Level Synthesis of Hardware Accelerators*. Thèse de doctorat, Ecole Normale Supérieure de Lyon, 2010.