

Sujet de stage de recherche – Master 2

Stratégies de compilation pour la synthèse de circuits

Encadrant: Christophe Alias, chargé de recherche à l'INRIA `Christophe.Alias@ens-lyon.fr`
avec la participation de Matthieu Moy `Matthieu.Moy@ens-lyon.fr`

Lieu: Laboratoire de l'Informatique du Parallélisme (LIP)
École Normale Supérieure de Lyon

Gratification: \approx 500 euros/mois.

Contexte

Depuis la limite de miniaturisation des transistors à densité énergétique constante, l'efficacité énergétique (calcul/J) est un problème central dès que le budget énergétique est limitée. C'est notamment le cas pour les systèmes embarqués et les calculateurs haute performance. Dans ce cas, la meilleure solution est de construire un circuit spécialisé. Depuis 2010, les FPGA se sont imposés dans le monde du big data et du calcul haute-performance comme une solution naturelle. Hélas, ils sont compliqués à programmer: il faut en effet construire, *from scratch*, un circuit pour la fonction à réaliser. D'où l'émergence de modèles de compilation de circuit (synthèse haut-niveau ou *high-level synthesis*, HLS) [3, 1] capables de traduire un programme C en une configuration de circuit FPGA.

Objectifs

Le but de ce stage est d'étudier les schémas de compilation possibles pour distribuer les données accédées en parallèle sur des bancs de stockage différents, tout en assurant la cohérence des données. Bien entendu, il n'est pas question de produire un circuit directement, on s'appuiera sur un outil de HLS existant ($C \xrightarrow{HLS} \text{circuit}$), qu'on nourrira avec un programme C optimisé par nos soins: $C \xrightarrow{\text{stage}} C \text{ optimisé} \xrightarrow{HLS} \text{circuit}$.

Plan de travail

- Etudier les caractéristiques de plusieurs outils de HLS (VivadoHLS, Altera/OpenCL, etc)
- A la main, optimiser quelques noyaux de calcul simples; évaluer plusieurs stratégies de génération de code.
- Implémenter la stratégie retenue. Tester et valider sur les benchmarks PolyBench/C [2].

Compétences souhaitées. Notions en compilation, en architecture des ordinateurs et en C++.

Références

- [1] Jason Cong, Bin Liu, Stephen Neuendorffer, Juanjo Noguera, Kees Vissers, and Zhiru Zhang. High-level synthesis for fpgas: From prototyping to deployment. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 30(4):473–491, 2011.
- [2] Louis-Noël Pouchet. Polybench: The polyhedral benchmark suite. *URL: <http://www.cs.ucla.edu/~pouchet/software/polybench/>*[cited July,], 2012.
- [3] Robert Schreiber, Shail Aditya, Scott Mahlke, Vinod Kathail, B Ramakrishna Rau, Darren Cronquist, and Mukund Sivaraman. Pico-npa: High-level synthesis of nonprogrammable hardware accelerators. *Journal of VLSI signal processing systems for signal, image and video technology*, 31(2):127–142, 2002.