Master of Science Internship Proposal
Circuit Synthesis and Modular Process Networks

Advisor: Christophe Alias, chargé de recherche at Inria, LIP/ENS-Lyon.
mail: Christophe.Alias@ens-lyon.fr
web: http://perso.ens-lyon.fr/christophe.alias

Duration: 4 – 6 months. This internship can potentially lead to a PhD thesis.

Place: Laboratoire de l’informatique du parallélisme, École normale supérieure de Lyon.
CNRS UMR 5668 – Inria – UCBL, web: http://www.ens-lyon.fr/LIP

Process networks are execution models expressing naturally the parallelism of a computation. This way, process networks are studied as models for parallel systems and allow to design theoretical studies, analysis, and measures for the various problematics related to parallelism. Process networks are a natural intermediate representation for parallelizing compilers, where the front-end extracts the parallelism and produces a process network and the back-end compiles the process network to the target architecture.

High-level circuit synthesis (HLS) consists in compiling a program written in a high-level language (as C) to a circuit. The circuit must be as efficient as possible, while using the available resources in the best fashion (consumption, silicon surface, FPGA LUT units, etc). Though many advances were achieved on the back-end aspects (pipeline construction, routing), the front-end aspects (I/O, parallelism) are still rudimentary and far less powerful than the approaches developed in the HPC community.

In that context, we have designed a model of process network that fits HLS-specific constraints. Our model makes explicit the communications with the central memory and the parallel access to channels, and is close enough to the hardware constraints to be translated directly to a circuit [2, 1].

This internship aims at studying how to extend our process network model with modularity. That is, the possibility to view a process network as a process in a bigger network, while keeping the efficiency of the global system. Such a feature would enlarge dramatically the class of programs currently handled, allowing bigger kernels and irregular constructions.

The student will:
• study the state of the art on automatic parallelization and process networks,
• invent a module system for our process networks, and propose an efficient compiler algorithm,
• implement, test and validate his approach on community benchmarks.

Prerequisites. Solid notions in compilers, computer architecture and parallelism.
References
