Laboratoire
d'Informatique
de Robotique
et de Microélectronique
de Montpellier

Master-level Internship (Sujet de Stage Recherche)
Static Analysis for Synchronous Dataflow Programs : temporal properties

2013

Location. This internship will take place in Lille. The collaborative work with Montpellier and Lyon will be done by means of conference calls.

Keywords. Static Analysis, Synchronous Programming, Code Generation, Compilation, Satisfiability Modulo Theory, Abstract Interpretation, Temporal logics.

Context. Addressing the correct design of embedded systems with multiple clock domains at the hardware level is very complex because of various factors, e.g., noise and jitter on clock signals, and skew between data signal and clock signals. As a solution, the problem can be addressed rather at a higher level.

The abstract clock notion provided by synchronous languages [1] offers the opportunity to suitably address the problem. An abstract clock is a discrete set of logical instants at which events are observed in a system, as illustrated in Fig. 1. Then, the synchronization relations between events of different nodes can be described as abstract clock relations. Thus, abstract clocks are very useful for the high-level design of multi-clocked embedded systems.

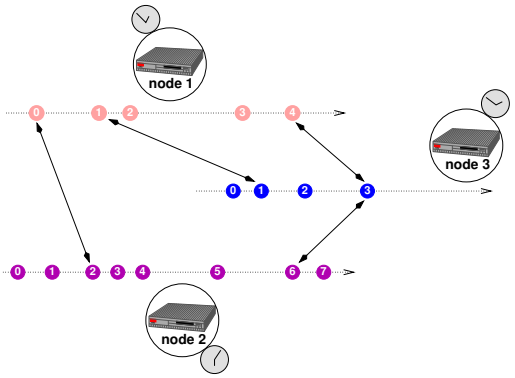

FIGURE 1 – A multi-clocked GALS (globally asynchronous locally synchronous system.

Currently, the static analysis of abstract clocks in programs, performed by the compilers of synchronous dataflow languages loses some relevant information when concerned clock properties include numerical expressions. As a result, this has a strong impact on the analysis precision and the quality of code generated automatically by these compilers.

Intership Subject. Recently in [2] and [3], we have designed a new static analysis for improving the code generation from synchronous multi-clocked programs, typically specified in Signal and addressed

with the associated compiler. Our analysis performs a better numerical and Boolean abstraction on the program variables.

We would like to develop this preliminary work in the field of software verification, and deal with temporal properties of synchronous dataflow programs.

The candidate student is expected to :

- Study the existing static analyses which have been designed in the synchronous programming community, in particular recent work using smt-based analyses.
- Propose an extension of the method proposed in [3] in order to deal with temporal properties and exploit the capacities of the state-of-the-art smt-solvers.
- Implement this extension inside our prototype and validate on examples from the litterature.

Desired knowledge / skills (but not limited). Background in formal/mathematics approaches (in particular, language semantics and logics) are required. The candidate will implement her/his method inside an existing static analyser written in OCaml.

Contact persons.

- Laure Gonnord, LIFL, University of Lille `Laure.Gonnord@lifl.fr`
- Abdoulaye Gamatié, CNRS/LIRMM – Montpellier `Abdoulaye.Gamatie@lirmm.fr`
- Paul Feautrier, LIP, University of Lyon `Paul.Feautrier@ens-lyon.fr`

Références

[1] A. Benveniste, P. Caspi, S. Edwards, N. Halbwachs, P. Le Guernic, and R. de Simone. The synchronous languages twelve years later. In *Special issue on Embedded Systems, IEEE*, 2003.

[2] Abdoulaye Gamatie and Laure Gonnord. Static analysis of synchronous programs in signal for efficient design of multi-clocked embedded systems. In *Proceedings of the 2011 SIGPLAN/SIGBED conference on Languages, compilers and tools for embedded systems*, LCTES '11, pages 71–80, New York, NY, USA, 2011. ACM.

[3] Paul Feautrier, Abdoulaye Gamatié, and Laure Gonnord. Enhancing the Compilation of Synchronous Dataflow Programs with a Combined Numerical-Boolean Abstraction. Technical report, LIFL, January 2013. Research Report.