Synchronous Languages for Hardware and Software Reactive Systems

Gerard Berry

Ecole des Mines de Paris

B.P. 207, F-06904 Sophia-Antipolis CDX, France, berry@cma.cma.fr

Abstract

Synchronous languages are dedicated to hardware, software, or mixed reactive systems that maintain a continuous interaction with their environment. They come in two classes: data-oriented languages such as Lustre and Signal, which are targeted to data-intensive applications such as continuous control or signal processing, and control-oriented languages or visual formalisms such as Esterel, Statecharts, or Argos, which are tailored for discrete controllers. Data-oriented languages are equational and involve combinational or delayed operations over multiply-clocked signals. Control-oriented languages have explicit primitives for sequencing, concurrency, and process preemption.

The synchronous languages are defined by precise mathematical semantics based on a simple zero-delay model, unlike most HDLs that are based on more intricate simulation models. The semantics makes it precise what a program means, how it can be compiled, and how to verify its properties.

We analyze the synchronous programming primitives and we present the semantical framework. We show how to synthesize RTL-level circuits from synchronous programs and how to use circuit synthesis and optimization techniques to produce efficient software codes as well. We present verification techniques and algorithms for synchronous programs. We discuss combinational cycles in synchronous programs, and we exactly characterize those that have a synchronous behavior. Finally, we present some industrial applications of synchronous languages.

More detailed information can be obtained from the Esterel web page at <http://www.inria.fr/meije/esterel> or by sending an email message to `<esterel-request@cma.inria.fr>`.

Keywords

Synchronous languages, reactive systems