

Lustre: A Declarative Language for Programming Synchronous Systems

Gregory Malecha

March 3, 2008

This paper serves as an introduction to the semantics of the *Lustre* language. Lustre is a language which is useful for reasoning about systems, and, in particular, about proving some types of properties of the system; though these are not discussed in the paper. This work is of interest to us because it can potentially be the basis for semantics for Uccello since it is able to deal with cyclic logic and clocks; two fundamental things for circuits.

The paper begins with an introduction to Lustre by describing the connections between constants and streams. Operators on streams are described as working piecewise on the elements in a stream and “memory” is introduced through the `pre` operator on streams which shifts streams forward so that the element being computed with is the value at the previous time step. Clocks are then introduced and the method for understanding computation as it is done in Lustre when working with different clocks is described.

The paper then presents a semantics for Lustre and defines the ways that this definition can be used to compile automata which can be used in verification. In addition, it presents some equivalences which can be used for optimization of compiled programs. Our interest in the paper lies in the clock semantics of value streams which are the primitives in Lustre.