An Implementation of Clock-Gating and Multi-Clocking in Esterel

Laurent Arditi, Gérard Berry, Marc Perreaut
Esterel Technologies
{Laurent.Arditi,Gerard.Berry,Marc.Perreaut}@esterel-technologies.com

Michael Kishinevsky
Intel Strategic CAD Labs
Michael.Kishinevsky@intel.com

Clock gating and multi-clocking are now common design techniques that are used for power reduction and for handling systems with different operational frequencies. They cannot be handled by Classic Esterel language and tools because the Classic Esterel is a single clock synchronous paradigm and Esterel compilers can generate single clock circuits only. To cover broader class of design needs, we propose to extend Esterel to other clocking schemes, including clock-gating and multi-clocking.

This extension must satisfy three major needs:

- enhance the scope of designs that can be modeled in Esterel,
- allows to generate different implementations depending on the final target: a single clock circuit (e.g. for compiling a specification into a basic FPGA), a circuit with clock-gating or an equivalent circuit without clock-gating, and a multi-clock circuit (e.g. for compiling to an ASIC). The choice of the implementation should be possible at compilation time, without requiring any change in the source model.
- provide support by all tools comprising the Esterel development framework: the graphical Esterel entry, software simulation and debug, embedded code generation, formal verification, optimization.

The core of the implementation for the clock-gating is based on a new Esterel instruction called `weak suspend`. This instruction freezes the control and data registers, while letting the combinational part computing the values as functions of inputs and the state. The effect of this instruction is similar to an effect of clock-gating on a hardware block. We developed an Esterel compiler which can generate RTL code (VHDL and Verilog) with the embedded clock-gating logic or with the regular equivalent logic to emulate functional behavior of clock-gating without the corresponding power saving.

The multi-clock design in the new Esterel compiler is based on the paradigm of Globally Asynchronous Locally Synchronous principle and is implemented using a few language extensions: multi-clock units, clock signals, clock gaters and clock multiplexers, and clock definition in module instantiations. The compiler can generate a truly modular and multi-clock RTL code, or mono-clock RTL code based on the `weak suspend` instruction. The latter compilation mode is also used for software simulation and formal verification. The trace equivalence of different forms of the design is guaranteed correct-by-construction.

We show a few classical multi-clock examples, including a dual-clock FIFO and a synchronizer based on a four-phase handshake protocol. A formal verification of this protocol is also discussed in detail.