Sujet de TER/stage, 2023

Study on the genericity of the Chisel language for the modeling and verification of hardware/software systems

Contact. Laurence Pierre, TIMA Lab.
Laurence.Pierre@univ-grenoble-alpes.fr

Context.

The implementation of hardware, or software/hardware, systems follows a *design flow* based on "*hardware description languages*" (HDL). They make it possible to describe (specify) the expected behavior of the system, analyze it in simulation, then proceed to the synthesis of the component. Ultimately, the final result may be the system-on-chip, or part of it.
Depending on the complexity and characteristics of the designed system, different types of HDL can be involved, allowing specifications at different "levels of abstraction" (roughly speaking, from a "transactional" description to a "logic gates" one, by way of behavioral descriptions, [1-3]). These languages generally offer *generic* description capabilities, one of the most typical cases is parameterization on the size of the component (example: arithmetic circuit for integers encoded on N bits).

To ease the design of very complex and highly configurable systems, "*hardware construction languages*" (HCL) have recently emerged. One of the flagship languages is **Chisel** [4], **based on the Scala programming language**. The objective in this case is to write Scala programs which will themselves generate HDL descriptions at the so-called "register transfer" level (data flow descriptions). One of the advantages is the possibility of configuring the system specifications on much richer aspects than sizes or delays for example (possibility to parameterize a behavior on conversion functions, communication protocol, etc.).

Goal.

Verifying the correctness of hardware or software/hardware systems obtained through a CAD process represents a major challenge, whatever the language(s) in use. There are various well-known methods, from verification in simulation to verification using formal methods, associated with HDL-based design flows.
The goal of the internship is to **analyze the possibilities offered by the use of Chisel to optimize the verification process**, in particular by taking advantage of its very high level of abstraction and its powerful genericity capability. One avenue here could aim to exploit automated proof techniques (see [5]), particularly those supporting higher order logic (tools HOL or Coq for example [6][7]).

The work will consist in:
- a study of the characteristics of the Chisel language, a state of the art of its main applications, and some experiments,
- a state of the art on the verification solutions proposed in the context of hardware construction based on Chisel,
- an in-depth look at Chisel's parameterization capabilities and their utility to design families of products,
- proposals to exploit these characteristics of Chisel for verification, in particular the higher order features. Depending on the state of progress, experiments could be carried out in an automated proof tool.

References.

[1] IEEE Std 1666-2005, IEEE Standard SystemC Language Reference Manual. IEEE, 2005.
[2] IEEE Std 1076-1993, IEEE Standard for VHDL Language Reference Manual. IEEE, 1993.
[3] IEEE Std 1364-2001, IEEE Standard for Verilog Hardware Description Language. IEEE, 2001.
[4] Chisel/FIRRTL Hardware Compiler Framework, https://www.chisel-lang.org/
[5] https://www.tptp.org/OverviewOfATP.html
[6] https://www.cl.cam.ac.uk/research/hvg/HOL/
[7] https://coq.inria.fr/