SuperTest – taking compiler risk out of RISC-V at Codasip

Codasip is a founding member of RISC-V International (formerly the RISC-V Foundation) and in 2015 became the first company to offer a commercial RISC-V IP core. Today, the company is a leading supplier of processor solutions for IC designers, offering products based on open standards such as the RISC-V ISA, LLVM and UVM. Applications for its standard and custom RISC-V cores range, among others, from audio processing, auto-focus camera systems and motor control, to neural networks, machine learning, and blockchain processing. The attractions of the RISC-V instruction set architecture are simple. It is modular, extensible, customizable via the inclusion of unique application-specific instructions, and there are no licensing charges for using it. That makes it ideal for accelerating applications at the hardware architecture level.

"If you discover a computational bottleneck in your application, with RISC-V you can create special instructions and hardware to deal with that bottleneck," says Roddy Urquhart, Codasip's Senior Marketing Director. "What we can do, or what our customers can do with our Codasip Studio tool, is implement those instructions, create a micro-architecture for the computation in hardware, and then automatically generate both the hardware description and the necessary software development kit, including the compiler, linker, assembler etc. at the touch of a button."

Presented with a custom ISA, Codasip Studio therefore generates a custom compiler. So how do Codasip's customers know the compiler works? To demonstrate their confidence in Codasip Studio's ability to automatically generate fully functioning compilers, the company turned to SuperTest.

"Our compilers are based on LLVM and use Clang as the front-end for C and C++, and that remains the same for all ISAs. Then Codasip Studio uses the CodAL processor description language to describe the unique instruction set of the processor. If you want to add a custom extension you describe it in CodAL, which is itself based on C," says Pavel Šnobl, Head of Codasip's Compiler Team. "What Codasip Studio then does is generate a custom compiler back-end to accommodate the new instructions, and on top of that adds a number of different code optimizations."

Prior to using SuperTest, Pavel and his team used standard open-source compiler tests suites but found them to be less than satisfactory.

"We originally developed our own test suite, which was based on the LLVM and GCC Torture test suites together with some of our own custom tests, but we couldn't really be sure that they covered everything," says Pavel. "It wasn't a very systematic test suite, it was more a selection of random tests."

One of the main reasons Codasip chose SuperTest was because it provided a much more systematic approach to compiler testing.

"With SuperTest, we can now reliably tell our customers that our compiler verification covers the entire C and C++ language standard. And it also gives us two other advantages. Firstly, SuperTest includes Solid Sand's CGTrainer code generator trainer, which provides us with a systematic way of testing the compiler back-end, the part that Codasip Studio generates to accommodate instruction set extensions. Secondly, it includes tests for the optimizations that we often develop for our customers in order to meet their performance or code size requirements. As a result, we can run tests that target every part of the compiler," says Pavel.

SuperTest also integrated easily into Codasip's development environment.

"Our testing procedure uses PyTest, which is a testing framework written in Python. SuperTest is also scripted in Python, so it was quite easy for us to integrate it into our existing test infrastructure. From our existing Python scripts we just invoke the SuperTest Python script and it runs," says Pavel. "In addition, the output of our current testing process is an XML report, which is also something SuperTest can produce, so the output reports from SuperTest can be easily integrated into our current test system. All these features made it very easy for us to get SuperTest up and running." Problem solved.

"Before, when customers asked us how do we really know our compilers work, we didn't always have a very good answer," says Pavel. "After using SuperTest, now we can say with a very high level of confidence that they do."

Codasip focuses on on-chip processor solutions for system on chip (SoC) designers. The offering includes the Codasip Studio processor design environment, the family of Codasip RISC-V embedded and application processor cores, and the SweRV Core™ Support Package for supporting open source RISC-V cores designed by Western Digital. Codasip solutions are based on open standards including the RISC-V open ISA, LLVM and UVM to ensure compatibility and longevity.

Solid Sands is based in Amsterdam, the Netherlands. Our mission is to put quality into C. We do that by improving the quality of C and C++ compilers, libraries and analysis tools, and their safe and secure use, with the best possible test and validation suite. With SuperTest, Solid Sands serves its customers to achieve the software quality level required by the ISO language and functional safety standards. With our history in compiler development, our knowledge of past, current and upcoming versions of the C and C++ standards, new analysis and optimization techniques and new use cases, Solid Sands stays at the fore-front of tools testing and validation.

SOLID SANDS
from Amsterdam is the one-stop shop for
C and C++ compiler and library testing,
validation and safety services.

Postbus 7897 | 1008 AB AMSTERDAM | The Netherlands | www.solidsands.nl