

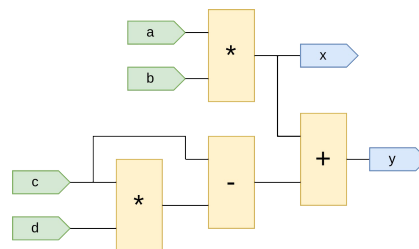
Designing a Hardware Testing Framework for SUS

Bachelor or Master Thesis

At a glance

- Learn about traditional hardware testing in SystemVerilog
- Research existing Hardware Testing Frameworks like cocotb
- Study Latency Counting in SUS
- Design and Implement the testing framework

At PC2 we currently develop a new language and compiler, named “SUS”, that aims to help in the design and development of high-frequency FPGA accelerators. It does this by explicitly encoding pipelining information in the type of each wire, and using it both to automatically balance pipelines, as well as participating in type inference of submodule parameters.



For testing and synthesis of the hardware that we design in SUS, we compile down to SystemVerilog. We then use testbenches written in plain SystemVerilog for testing. An issue we currently face with this approach is that traditional SystemVerilog has no notion of “Latency Counting” as it is used in SUS. This forces the written SUS code to explicitly declare Absolute Latencies on the interface, as well as the force test code to handle the pipeline stages that the SUS code implicitly introduced.

With the design of a new framework, possibly built upon cocotb, possibly written from scratch, we can take into account the paradigm under which SUS works, and handle the latency information of the module gracefully. This will make testing of SUS code easier and less error-prone.

Recommended Background:

- Understanding of hardware design
- Software development experience

Further reading:

<https://github.com/pc2/sus-compiler>

<https://docs.cocotb.org/en/stable/>

Contact:

Lennart Van Hirtum, Phone: 05251/60-4542 E-Mail: lennartv@mail.uni-paderborn.de