

Boost your productivity in FPGA & ASIC design and verification

Tutorial abstract DVCon Europe 2022

Most EDA tools take your HDL code as a starting point and process it as efficiently as possible, focusing on the silicon. Those tools do not care how you come up with the code in the first place. However, the way you come up with the code might be especially important in domains like functional safety development.

If you want to be more productive as a hardware design or verification engineer working on VHDL, Verilog and SystemVerilog code, you deserve a software tool that helps you to create and explore code in a more efficient way. Software development has a lot of amazing tools and the lessons learned from designing complex software systems also apply to designing complex hardware systems. So for hardware development you'll benefit from using similar tools.

In this tutorial, Bart Brosens (Application Engineering Lead at Sigasi) will demonstrate how to increase your productivity using an IDE for your HDL design. You'll learn our best practices from proven methodologies such as:

- type-time feedback,
- intelligent content assist,
- various ways to explore and navigate through large projects,
- how to effortlessly document your design
- and much more.

Some of these best practices are extra appealing when you're working on projects following functional safety standards like DO-254 (aerospace) and ISO 26262 (automotive). For example, when you review code, you want to be able to focus on the functionality of the code, rather than cosmetic aspects of the code. You'll be able to focus on reviewing the functionality of your code much faster if this code was created using a tool that checks for naming convention violations and takes care of a consistent formatting of the HDL code.