New UNISIM Libraries for Functional VHDL

With the new UNISIM libraries from Xilinx, you can simulate RTL behavioral code with gate-level instantiations, gate-level descriptions imported from schematics, and gate-level VHDL and Verilog descriptions exported from synthesis, prior to place and route.

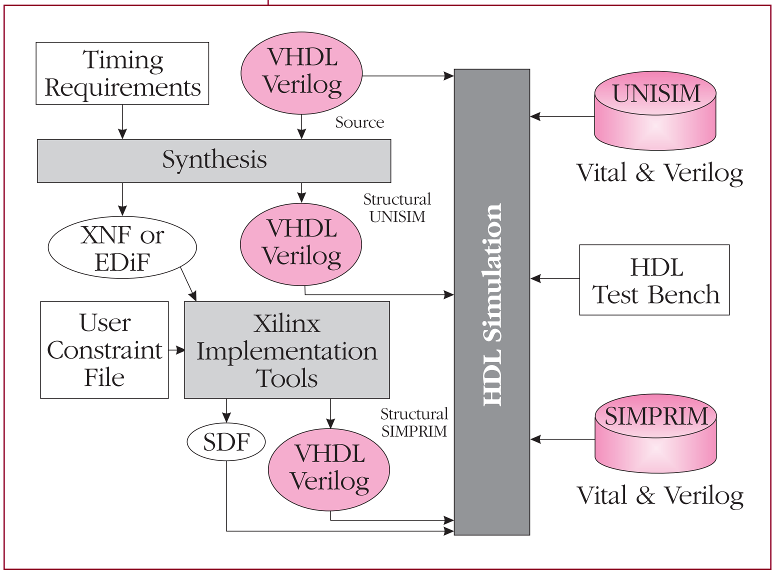
> *“The libraries are tailored for synthesis-based HDL design flows and include special HDL support for global signals.”*

These new libraries complement the VHDL and Verilog SIMPRIM Libraries currently available for timing simulation, and are available in the Alliance Series and Foundation Series 1.4 software, completing the Xilinx HDL-based flow. **Figure 1** illustrates the new simulation flows that are now possible.

The libraries are tailored for synthesis-based HDL design flows and include special HDL support for global signals. They are called UNISIM to distinguish them from Unified Library schematic-based libraries, and contain all the Unified Library cells used by synthesis, as well as the cells that are instantiated due to limitations in inferencing.

You can now instantiate cells such as I/Os, RAMs, ROMS, oscillators, and so on, in their RTL code, and proceed to simulation without converting your synthesized design to SIMPRIMS. Furthermore, the UNISIM Libraries are fully compatible with the LogiBlox-generated behavioral models. Designs can have both LogiBlox and UNISIM instantiations. Alliance synthesis vendors are planning to write structural VHDL and Verilog that is compatible with the UNISIM libraries to enable the post-synthesis simulations that are particularly useful for verifying high-density synthesis results prior to implementation.

Figure 1 - *UNISIM libraries.*



Because Verilog lacks a configuration statement to choose between different behavioral models, a Verilog library has been created for each Xilinx technology. VHDL, on the other hand, does have a configuration statement, allowing you to select between different models for the same component. Therefore only one VHDL library has been created for all Xilinx technologies.

Whether the pre-route descriptions are VHDL or Verilog, special attention must be paid to global signals in HDL code. Both the VHDL and Verilog UNISIM libraries have mechanisms for handling GSR, GR, PRLD, and GTS signals to match pre-route and post-route initializations. The mechanisms are adapted to use the features that are best suited for each language.

The Verilog testbench can be used to drive internal signals. Therefore, the UNISIM library uses Verilog macros to define global signals. You set variables to drive the GSR, GR, PRLD, or GTS global signals used in the macros. After the variables are defined, the macros are activated and the reset and configuration 3-state conditions are enabled.

VHDL in contrast, requires a port for every signal to be stimulated by a testbench. Furthermore, it also does not have a good mechanism for handling global signals in a way that is VITAL compliant. Therefore the following five unique cells have been developed to support the VHDL synthesis and simulation design flow:

and Verilog Simulations

- **ROC**: Emulates the chip-generated *reset on configuration* pulse.
- **ROCBUF**: Allows the testbench to drive the chip-generated *reset on configuration* without implementing an actual input pin on the chip.
- **TOC**: Emulates the chip-generated *3-state on configuration* pulse.
- **TOCBUF**: Allows the testbench to drive the chip-generated *3-state on configuration* without implementing an actual input pin on the chip.
- **STARTBUF**: A technology-independent version of the STARTUP block supported for simulation.

These five cells allow control over the global reset and 3-state signal emulation so you can create pre-route initialization simulations to match post-route simulations. The cells also drive implementation tools to add or delete pins, and to select which net is to be routed as the GSR, GR, PRLD, or GTS net.

Summary

The UNISIM libraries for Verilog and VHDL give you an HDL flow for RTL with instantiations, imported schematics, and post-synthesis functional simulations. The libraries are completely compatible with synthesis- and LogiBlox-oriented flows and will help you match your pre-route global signal initializations with your post-route simulations. ◆