

# Re-thinking Your Verification Strategies for Multimillion-gate FPGAs.

How do you alter your verification techniques to meet today's high gate count requirements? It depends on your background and experience.

by Thomas D. Tessier

President, f2design Incorporated  
tomt@hdl-design.com

FPGA verification is essential for successful on-time product delivery, and today's million-gate FPGAs require you to re-think your old verification strategies. Many engineers continue to use simulator-specific approaches for verification; the simulation tools are primarily used for module testing, while the lab is used for system-level integration. This approach requires the engineer to manually stimu-

late signals and view the resulting waveform responses. Because this process is time consuming, error prone, and difficult to repeat, engineers often spend minimal time in simulation and quickly move to debugging in the lab. Multimillion-gate FPGAs implement functions far too complex to rely on this ad-hoc method.

Designers are choosing million-gate FPGAs because they are fast enough and

large enough to handle the design complexity that was previously achievable only with an ASIC. When ASIC engineers begin to use high density FPGAs, they take their verification approaches with them. Those who use a validation process with robust tools and a complete self-checking testbench environment find that continuing to use their familiar testing approaches now causes them to lose valuable design cycle time. ASIC

Designers can benefit from a carefully defined and executed verification plan that takes FPGA reprogrammability into consideration. Time that was once well spent in exhaustive verification at the RTL level with an ASIC, now becomes costly for a high density FPGA.

**What is Verification?**

Verification is not synonymous with simulation. It is a strategy to make sure all parts of the system conform to the specification document; simulation is a tool used in the verification effort. The basic components of verification are shown in Figure 1.

**Specification**

A detailed and complete specification is essential for producing working products, on schedule. The specification document is the foundation of the verification plan, and describes the features to be implemented, under what conditions they occur, and what their expected outputs should be. This documentation should not determine implementation—that is left to the experience of the RTL designers.

**Verification Plan**

RTL engineers and verification engineers share the responsibility for implementing the test plan. The level of test granularity (or detail) is outlined at: transactions, protocol, interfaces and timing. Essential functions are identified. A determination of the number of testbenches needed, their complexity, and test module dependencies is made.

Any discrepancies in design implementation versus testbench results should be referred back to the specification for clarification. This is not a new concept but often overlooked in the rush to produce a product. When all elements described within the test plan are checked off, the verification effort has been completed to the required level of confidence. To optimize your verification effort the following

list offers examples of the type of information you need to identify:

- External interfaces
  - Stimulus and response
  - Transaction level, such as Read vs. Write operations
  - Timing requirements
- HDL models available to assist in testbench development
  - Packaged with proposed Intellectual Property (IP)
- Tools available to the project
  - Simulators
  - Static Analysis
  - Lab-based tools
- Performance Requirements, such as: need 32 block data write @ 66 MHz with a latency of less than 300 ns.

**Execution**

A verification strategy that best suits your design means breaking out those func-

tions that are essential to simulate and those that can be tested during in-system test. The execution of the Verification Plan requires simulation and in-system test on the target PC board—the final stages of the pyramid.

**Verification Simulation**

Simulation has two components:

- Dynamic simulation describes behavioral HDL, RTL, and gates.
- Static analysis encompasses Static Timing Analysis (STA), Formal Verification and Signal Integrity Analysis.

**In-System Test**

During in-system test you have a distinct advantage when using FPGAs over ASICs. An obvious benefit is the ability to reprogram the FPGA until the desired functionality is achieved. You also have an additional advantage with the Xilinx ChipScope Integrated Logic Analyzer which enables you to observe internal nodes of the chip, on your PC board, while running at system speeds.

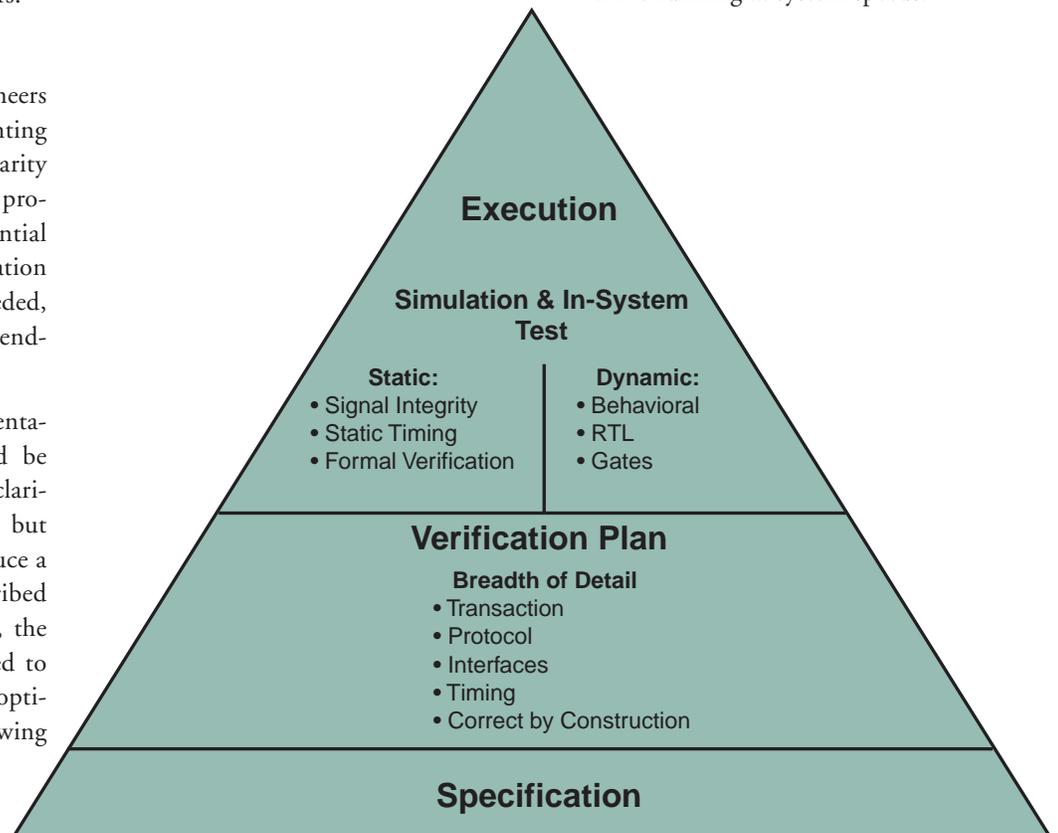


Figure 1 - Verification pyramid

## Interaction of Verification and Design Creation

Verification has many interactions with design creation, as shown in Figure 2. To prevent confusion and save time, the design and verification teams must work from the same thorough specification. In addition, the RTL design engineers and verification engineers must share the responsibility for implementing the test plan—testbenches are written to validate the design to the specification, not to verify the design implementation.

Once the executable specification (of the design) and testbench, both written in behavioral HDL, meet the requirements, the design is replaced with RTL code. The RTL is then verified with the system-level testbenches to make sure it meets the written specification conditions. After the RTL is validated it is synthesized and processed by the Place and Route tools. The resulting gates are plugged into the system verification testbenches or formal verification if it is available. This insures the tools have correctly implemented the design. In addition generated gates are run thorough static timing analysis. This step verifies that the system-level timing is met.

System integration is typically referred to as “power on”. This is the time when project teams come up with creative answers to the question “is it working yet?” Projects are ready for in-system test when they have validated RTL code, have been successfully placed and routed, and can create a bit stream to program the FPGA on the physical PCB. At this point it is expected that module-level partitions have been tested for functionality, and that module interfaces are stable and well defined. The design has been simulated, as a chip, at both RTL and gates levels with minimum functionality necessary for power on. The simulation of the chip is often not achieved by FPGA design teams still using simulator specific approaches.

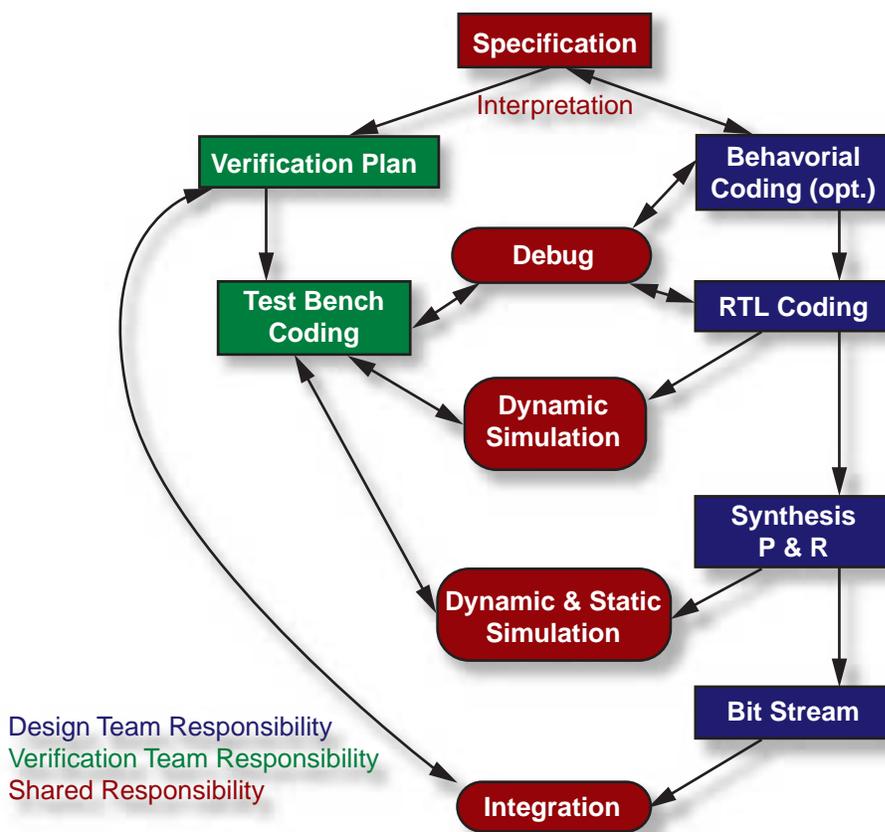


Figure 2 - Interaction of the verification components

## Conclusion

A verification strategy combining simulation, static analysis, and in-system testing is key to success with high density FPGAs. You are bombarded with many different choices for verification of a design; to meet time-to-market pressures you need to leverage multiple approaches.

A detailed application note is available to guide you through the verification decision process, including an in-depth case study. It evaluates the design-specific trade-offs of choosing functions that are essential to simulate and those that can be tested during integration. Prepackaged IP testbenches are also evaluated for applicability in the system testbench. The full application note can be found at: [www.hdl-design.com](http://www.hdl-design.com).

## About t2design

t2design, Inc. provides HDL design and methodology process management solutions. We specialize in customizing project methodologies that create a cohesive, re-creatable design process from the architecture phase through verification. High density FPGAs, such as the Xilinx Virtex and Virtex-II families, require an "ASIC like" HDL design approach which means HDL code, simulation, and synthesis. Data flow and process management planning need to accompany the HDL design approach to create a complete project solution. Our team of designers implement this ASIC and FPGA methodology strategy while leveraging EDA tools to their fullest level of effectiveness. Contact us at 303-665-6402 regarding t2design services.