Verification *for Higher Productivity*

We take you to the leaders.

by Hitesh Patel and Carol Fields, Xilinx Alliance Marketing, hiteshp@xilinx.com, carol@xilinx.com

To maintain competitiveness, many designers have found that high-level Hardware Description Languages (HDLs) are essential for representing and verifying their designs. ASIC designers adopted the HDL design and verification methodology several years ago, because it works at a higher level of abstraction, allowing them to produce more gates per day.

With the introduction of the Xilinx Spartan family in 1997, along with the Xilinx mask programmed HardWire capability, FPGAs are rapidly becoming Gate Array replacements, and FPGA designers are now looking for a verification strategy to help improve productivity.

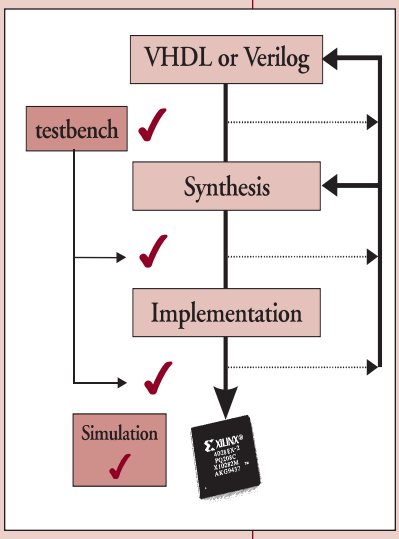

Figure 1: HDL Verification Design Flow

Verification Methodology

As design density and complexity increases, the cost of correcting errors increases exponentially with time. An error, if detected early in the design cycle (during RTL simulation), is fairly inexpensive to fix. That same error, if caught late in the design cycle, may necessitate a redesign and re-verification. Furthermore, debugging a finished device is very time consuming and often impossible because the debugging tools may not allow you to observe all internal nodes.

Perhaps the most publicized example of a costly bug was the one found in Intel's Pentium processor, which lead to Intel announcing a $475 million loss against fourth quarter earnings in 1994. To avoid such costly mistakes, designers are simulating their designs using a testbench methodology that allows them to observe all internal nodes and isolate errors early in the design process.

The design and verification methodology, shown in **Figure 1**, consists of four major steps: design entry, synthesis, implementation, and verification. You will often need to move between these four steps to either correct or change the circuit, and that requires you to verify the design at the following stages:

- RTL functional simulation
- Post-synthesis simulation
- Post-layout static timing analysis
- Post-layout simulation

Using a Testbench Methodology

A testbench is a separate set of VHDL or Verilog code that you use to specify circuit input stimuli and output responses, then you test to that specification at various points in your design cycle. This allows you to readily identify and resolve problems early in the design process, thereby saving significant time and costs. Steve Winkelman of DisplayTech states "I reduced my design cycle by 25% by adopting an HDL simulation methodology" *(see **Figure 2**)*.

A testbench also allows you to perform hardware regression, software debug, and system debug, in parallel. The same testbench (and simulator) are used before implementation to catch timing errors, and after synthesis to catch critical path problems (with estimated logic block and net delays).

Many FPGA designers are constantly pushing to higher density FPGAs such as the XC40250XV (500k system gates). The number of vectors required to verify these designs has risen many times faster than the size of the devices themselves, as shown in **Figure 3**. As a result, the amount of time spent in simulation increases dramatically. To help verify more in less time, other verification tools and strategies similar to ASIC strategies are being adopted. These tools include static timing, formal verification, cycle

XILINX®

base simulation, and emulation. They significantly speed-up the verification cycle times as opposed to simulation, which can take a long time and consume large amounts of memory.

The Changing Role of Verification

A design for verification strategy focuses simulation at the RTL level where it runs the fastest. The strategy replaces gate-level simulation with equivalence checking, which does a complete job of verifying that the low-level logic matches the RTL specification in a fraction of the time needed to simulate a meaningful vector set. With this strategy, only netlists that have already been proven equivalent to the "golden" RTL code are checked for timing requirements. As a result, timing issues are separated from functional issues and the timing simulation can focus solely on timing issues.

Emulation, used in conjunction with equivalence checking, allows you to run real applications with the certainty that these diagnostics also check the corresponding RTL code. Today, system designers and their verification counterparts can emulate large sections or even whole systems using the high capacity, high performance Xilinx FPGAs currently available.

Formal equivalence checking replaces regression simulation of gate-level implementations. Once an RTL specification is signed off, it serves as the reference against which implementations are compared. The fully verified implementation serves as the reference for later revisions. These formal comparisons provide complete coverage without vectors.

Why Should You Choose Xilinx?

Xilinx , from the very beginning, has used the industry standards such as VITAL, VHDL, Verilog, and SDF. These standards were developed for ASICs and have been applied to FPGAs, because as FPGAs replace ASICs, these standards are becoming critical for seamlessly integrating FPGAs and CPLDs into your existing flows.

The Xilinx AllianceEDA program and the Alliance Series software team provides the highest quality support for the industry-leading HDL simulators and emerging verification tools. By closely monitoring industry trends, we are prepared to meet your changing requirements. In the Alliance Series 1.5 release you will achieve the highest quality results. You will also find minimum delay characterization for hold time and race condition checking, Global Set-Reset (GSR) simulation models that depict the true behavior of the silicon, and a robust set of documentation supporting the industry's leading HDL simulators.

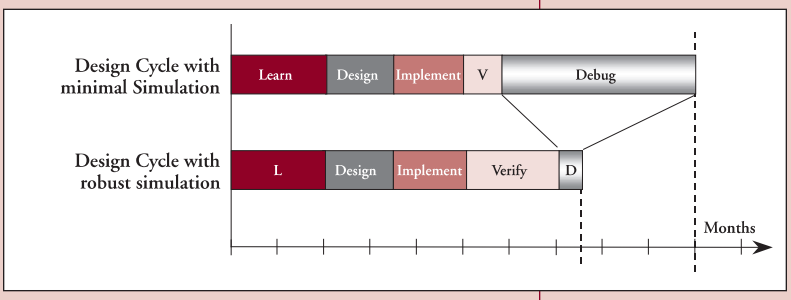

Figure 2: Productivity increases when using HDL simulation

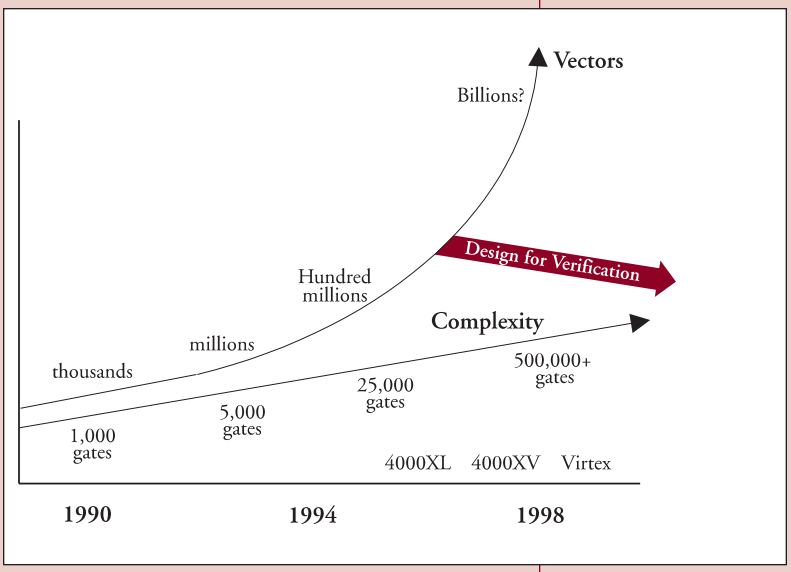

Figure 3: Vector requirements as gate densities increase.

Summary

Design verification is arguably the most critical task in successfully creating complex designs and decreasing time to market. Without verification, error isolation is a very tedious and time intensive effort.

In practice, engineers have found that a verification strategy improves their productivity and keeps design projects on schedule. As devices surpass one million gates, a design for verification strategy provides you with the confidence that you are releasing functionally correct devices.

Xilinx will continue to lead the way in developing advanced methodologies with our premier EDA partners. ◆