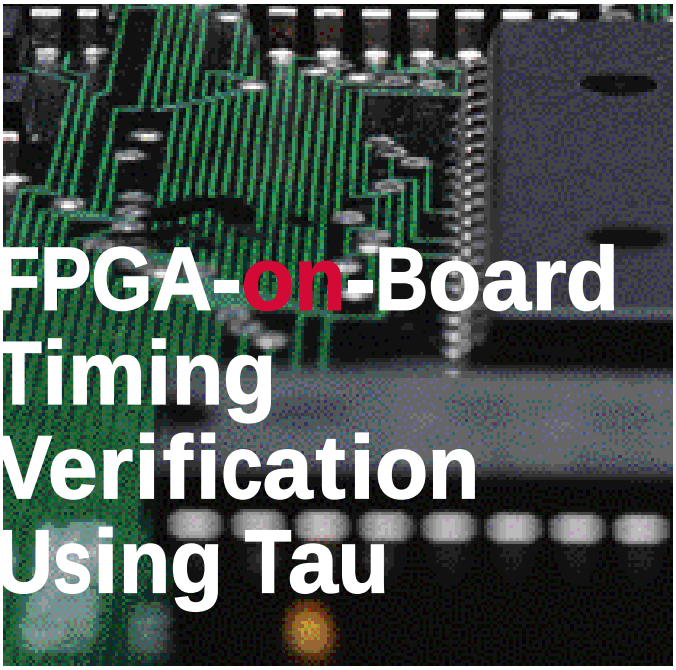

FPGA-on-Board Timing Verification Using Tau

by Ajay J. Daga, Tau Product Line Manager, Mentor Graphics Corporation, ajay_daga@mentor.com

Timing verification is increasingly becoming an important step in board design flows. With clock speeds frequently exceeding 50 MHz, timing margins have become progressively tighter. When you factor in the impact of clock skew, interconnect delay, and the delay through custom logic (FPGAs, for example), it is no longer possible to rely on Excel spreadsheets, back-of-the-envelope calculations, or stress tests of prototypes, to ensure that board designs will work reliably, at intended speeds, under worst-case conditions. Indeed, timing verification is now the fastest and most thorough approach to finding problems early in your design cycle.

Roger Yang, senior verification engineer at Cisco Systems says, "Timing verification is a very important step in our board design flow and we use Tau™ for this purpose. We find that an automated solution to board timing analysis is essential to fully, and in a timely manner, consider the impact on circuit timing of issues such as clock skew, interconnect delay, and delay through FPGAs and ASICs."

Tau, from Mentor Graphics, is a board timing verification solution. Tau takes as input a netlist, models for components in the netlist, and interconnect delay information (Figure 1). Tau uses this information to exhaustively, and in one pass, verify all timing constraints (set-up/hold times, or pulse-width times) on a design. Tau reports timing margins for each constraint and the skew between clocks, taking only a few seconds to completely analyze most board designs.

To illustrate the board timing verification process, consider the example circuit in Figure 2. It contains an IDT79RV5000 processor communicating with memory through a custom memory controller implemented using a Xilinx FPGA. The processor/memory controller interface is synchronous and is driven by a 100-MHz clock that is generated by the processor. The memory interface is asynchronous to the processor interface and is driven by a 40-MHz clock on the memory controller. The netlist for the design is provided to Tau either from a schematic capture tool such as Design Architect (from Mentor Graphics) or through EDIF.

Board Timing Verification with Tau

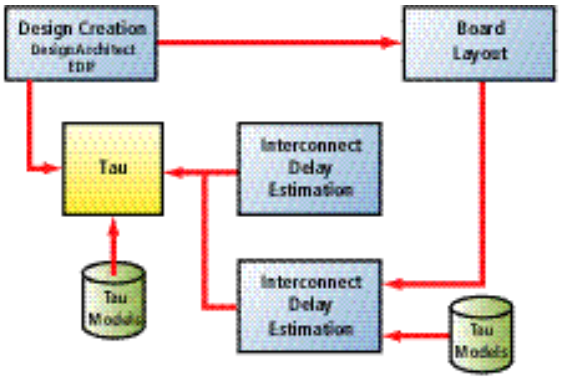

Figure 1

Models

Tau models provide black-box timing information for a component (pin-to-pin delays and constraints). Additionally, you can attach timing diagrams to a Tau model that describe read/write cycles. Tau uses the information in a timing diagram to automatically eliminate the reporting of false timing violations (for example, checking set-up/hold times every clock

(Continued)

Example Board Design

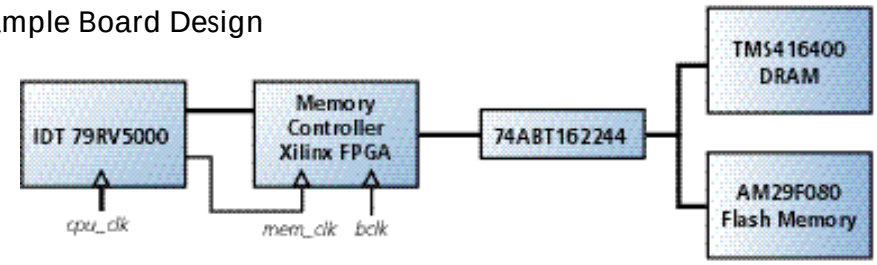

Figure 2

cycle when data is really latched at every other clock edge).

Models for the off-the-shelf components (IDT79RV5000, 74ABT162244, AM29F080, and TMS416400) on a design are obtained either from the Tau Model Library provided by Mentor Graphics (at no charge), or through Mentor's Tau project modeling service. This service provides models, with a two-week turnaround time for all off-the-shelf components on a design, for a fee of $3,000.

To model custom components, such as the Xilinx FPGA in Figure 2, Tau interfaces with model formats. Xilinx provides the ability to generate black-box timing information for FPGAs in the Stamp format. Stamp files are directly imported into Tau and design engineers do not have to manually re-enter this information. This is particularly important given the many revisions an FPGA goes through and the need to have immediate access to up-to-date FPGA timing information for accurate FPGA-on-board timing verification.

Timing Diagrams are a Source of Component models

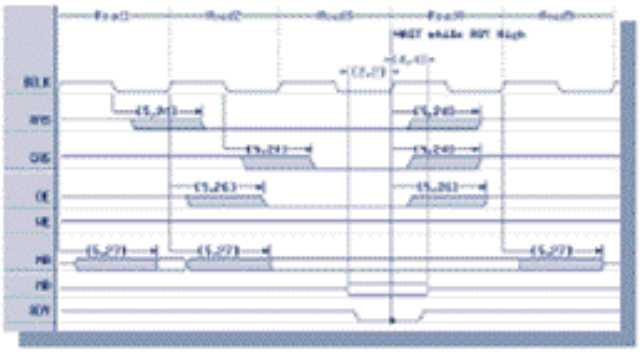

Figure 3

Finally, to prevent Tau from reporting false timing violations, you can import timing diagrams drawn using Chronology's TimingDesigner (such as that shown in Figure 3) that describes the read access on the memory controller. Timing diagrams capture the sequence of clock cycles that constitute a memory access. They specify when outputs change, when inputs are latched, and the values on control signals. Timing diagrams are useful for accurate analysis of memory subsystems because they capture phase and multi-cycle relationships between signals on an asynchronous interface.

Interconnect Delay

Interconnect delay is imported into Tau using one of two approaches. The first approach is to estimate interconnect delay based on placement information and a user-provided value for interconnect delay as a function of length. This approach is useful early in the design process when a board has been placed but not routed. It provides a "ball-park" estimate for interconnect delay and helps focus your attention on the potential problem areas of your design.

For a more thorough and accurate characterization of interconnect delay, you can use transmission-line simulation tools such as IS_Analyzer™ (from Mentor Graphics) or XTK™ (from Viewlogic). These tools use IBIS models to accurately characterize interconnect delay under different "corner" conditions while taking into account the impact of crosstalk, net topology, reflected-wave switching, and so on. Transmission line analysis is traditionally performed when a design is routed and all the physical information is in place for accurate delay characterization.

(Continued)

Report of Timing Margins

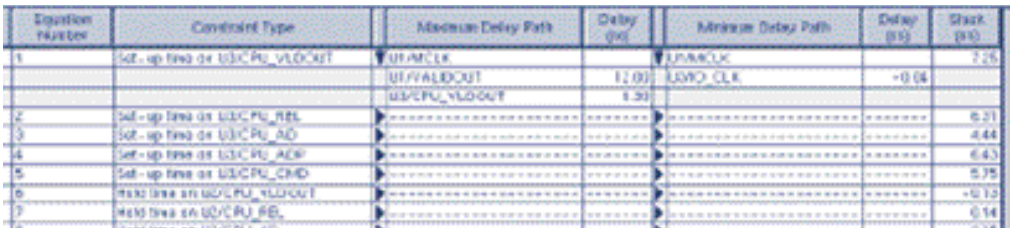

Figure 4

Timing Verification

In addition to a netlist, component models, and interconnect delay information, Tau requires a description of the clocks on a design. You specify the frequency, duty cycle, and peak-to-peak jitter at the source of a clock tree and this information is propagated to the clock nets driven by the source. So, for the design in Figure 2, a clock frequency of 100 MHz, with a 50% duty cycle and 250ps of peak-to-peak jitter, is defined at CPU_CLK. Tau automatically propagates this information to MEM_CLK. Asynchronous clocks are easily specified by allowing multiple clock sources on a design. So, in Figure 2, BCLK (used to drive the memory interface) is also a source clock.

Worst-case timing verification of the entire design is performed in one analysis run. Tau computes the skew between clocks and does so taking into account inversion in the clock tree and output skew (skew between output pins) on a component. Correlation in delay between clock and data paths is also taken into account. For example, in Figure 2, the IDT79RV5000 generates both clock and data signals for the memory controller. If the delays within the IDT79RV5000 track, as they probably will, then this is captured in the Tau model for the component in either percentage terms or picoseconds of skew.

Tau reports timing margins, or slack times, for each constraint verified, as shown in Figure 4. Negative slack indicates that a constraint is violated. Each constraint can also be viewed in block diagram form, as shown in Figure 5, to see the propagation paths for the constraint and component/interconnect delays for each segment in these paths. The block diagram view is particularly useful in helping isolate and fix timing problems.

Through the Stamp interface to Xilinx, the timing information for each revision of the memory controller FPGA is imported into Tau and the impact on timing margins is quickly and easily established. This is important because the memory controller timing is key to the overall board timing;

Block diagram view of the paths for a timing constraint

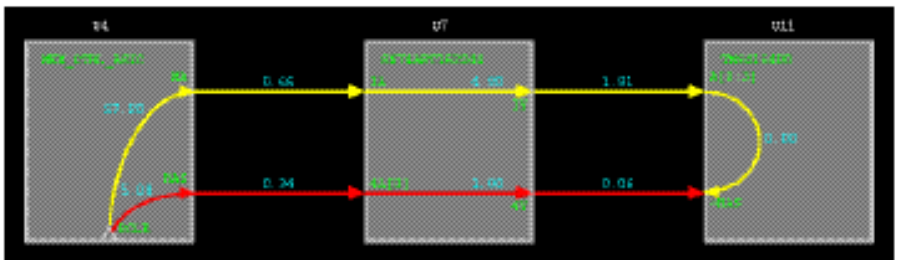

Figure 5

the controller communicates with the IDT processor at 100MHz and also drives an asynchronous interface to DRAM and Flash Memory.

Summary

Increasing clock speeds, the growing impact of interconnect delay, and the frequent use of FPGAs in the timing-critical portion of a design, necessitate board timing verification. There really is no other alternative to ensure the mass production of reliable boards that work at intended speeds under worst-case conditions. The Stamp interface from Xilinx, in conjunction with Tau, provides you with a viable and easy-to-use solution.