WP174 (v1.0) March 13, 2003

Methodologies for Efficient FPGA Integration into PCBs

By: Philippe Garrault

Since their invention, FPGAs have continuously and rapidly expanded in their capabilities. Evidence of this growth is abundant whether one looks at the core programmable logic that went from tens to millions of gates with added arithmetics, memory, clocking, and processing functions or at the I/O ring that has gone from a few I/Os to over a thousand, supporting numerous single-ended and differential standards with on-chip terminations and data rates into the gigabit per second range. The combination of rich feature set with device programmability and re-programmability has made FPGAs highly adaptive to design changes *during* and *after* product development and, consequently, is very attractive to a large variety of applications. For these reasons, there is growing acceptance of FPGAs as central components in system design. With this in mind, the following pages present a set of methodologies for efficient PCB design involving FPGAs.

This document helps readers to grasp how PCB design considerations play a major role in obtaining the expected performance from FPGAs. Specifically, it focuses on early analysis and simulation methodologies as a way of performing a guided implementation. That is, if variables affecting the design under development are analyzed and results passed onto the implementation tool, then it is more likely the desired design specifications will be met in the first implementation pass, fulfilling the ultimate goal to keep development effort, cost, and time to a minimum.

Like any other large IC on a board, FPGAs require attention to the implementation of their power delivery, cooling, and I/O systems. However, unlike other ICs, FPGA internal logic and I/Os are totally configurable by designers. This creates a broad range of parameters defining the FPGA's behavior and requirements. To guide FPGA board implementation, vendors always provide guidelines and documentation. However, as FPGAs offer ever-increasing leading-edge performance and application breadth, it becomes a challenge to provide the developers with meaningful numbers for these parameters. Device behavior and requirements intimately depend on the configuration of its internal logic, external I/Os, and the interaction of other components on the board. Since vendors do not have control over these parameters, they offer methodologies and models that allow designers to perform analyses specifically tailored to their application.

The first task considered in this document assumes that (1) the entire system has been divided into subsystems, a subsystem being a single PCB, and (2) the decision has been made to implement part of the system functionality into one or several FPGAs. Specific analyses related to FPGA integration are highlighted on the PCB design process flow chart (Figure 1). The following sections discuss in greater detail the issues affecting FPGA behavior and present techniques to handle them.

FPGA Selection and Design Planning

After deciding to implement part of the system functionalities into the FPGAs, designers and managers select, for each FPGA, the vendor, the family, and the particular device within the family. They evaluate the feasibility of the design in a particular family and list the resources that would allow the system to meet its requirements. At this stage, it is also appropriate to plan and allocate time and resources for the different tasks necessary to implement an FPGA into a PCB.

For the latest data sheets, handbooks, and application notes related to a Xilinx FPGA family, refer to the website: **http://www.xilinx.com/support/library.htm**
These resources provide insight into FPGA features and behavioral, electrical, and physical characteristics. When evaluating the feasibility of a design, another source of information for the designer is to contact a sales representative or the application engineers, who can help identify the FPGA resources needed and provide helpful information about setting up project planning.

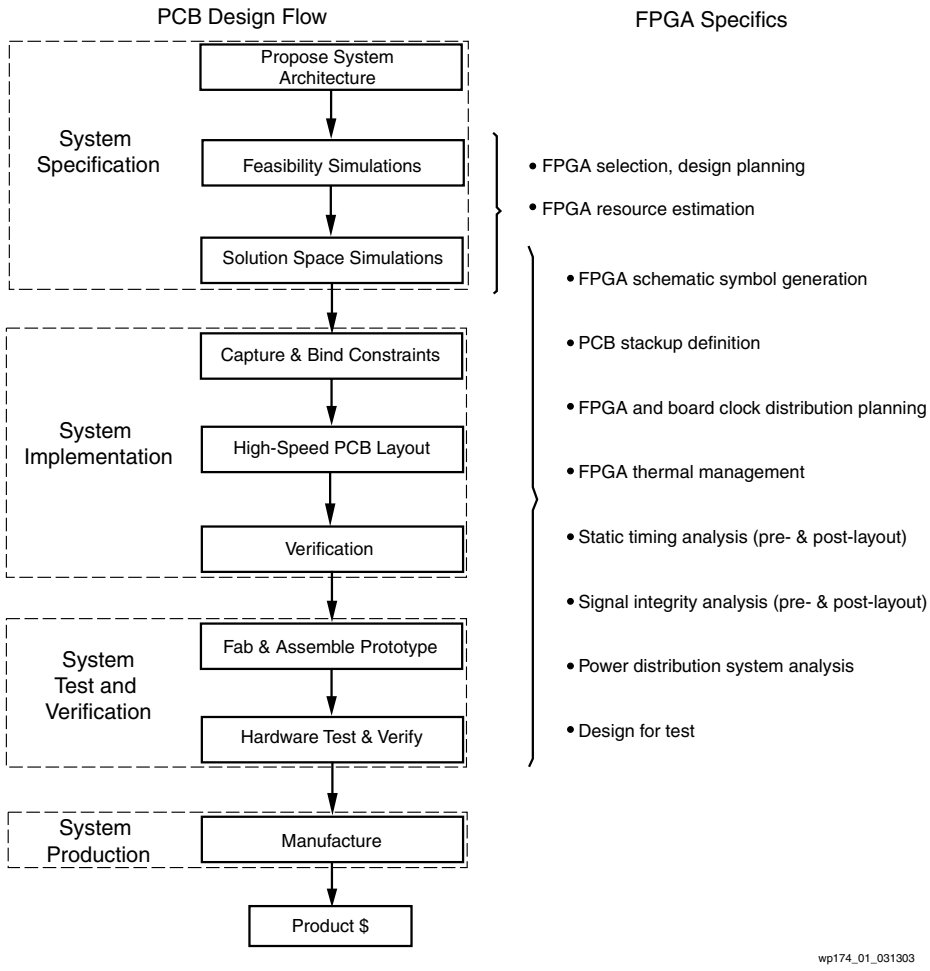

Figure 1: **High-Speed PCB Design Flow and FPGA Specifics**

FPGA Resource Estimation

After a device has been selected, attention should be given to evaluate in greater detail the system-level FPGA requirements. It is necessary to estimate both the FPGA requirements on the PCB, as well as any other supporting component required for proper FPGA behavior.

The following actions are recommended:

- List the different I/Os. Investigate the different communication channels between the FPGA and its environment to reveal the number of links and their physical and electrical characteristics.
- Plan the FPGA I/O partition:
  - From the FPGA side, the pinout affects internal routes to and from the FPGA I/O. In addition, there are electrical constraints referred to as I/O banking rules that can restrict pin assignments, for instance, constraints like compatible I/O standards, simultaneous switching output (SSO), etc. These limitations are well characterized and published by the FPGA vendor.
  - From the board layout side, the routability of hundreds of signals over a limited area can make the board engineers' tasks (e.g., IC pin assignment, symbol generation, or board placement) very tedious and time consuming. However, tools are available to help make decisions in these areas. For

example, Figure 2 shows a tool (from Product Acceleration, Inc.) that optimizes pin assignment and IC placement on a board to reduce wire crossing.

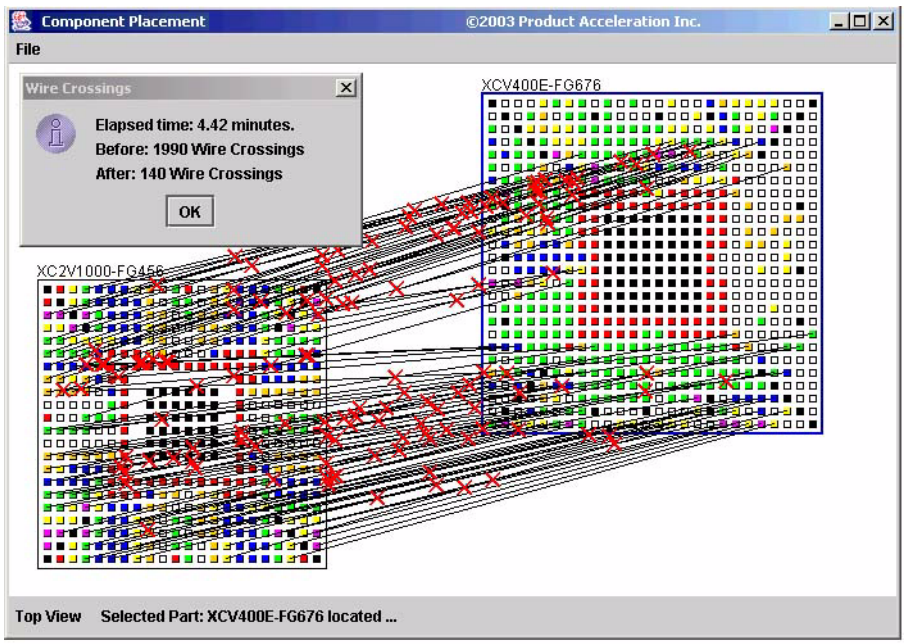

Figure 2: **FPGA Placement and Pin Assignment Optimization in DesignF/X™**

- Estimate the clock signal generation and routing at the board level. Because such signal routes are continuously switching, they can be a source of crosstalk and can also create electromagnetic interferences (EMI) if not terminated properly. On the board side, different layout techniques such as stripline routing help minimize these effects. Clock timing integrity is usually an important factor. For this reason, clock routing deserves special attention during the board-level signal simulation phase. On the FPGA side, taking advantage of the advanced embedded clock generation modules (digital clock management (DCM)) can help minimize the number of component and clock signals on the board and reduce the frequency of the board clock nets.
- Estimate the number of distinct FPGA DC power supplies. List the different supply and reference voltages needed for the core, the FPGA I/Os, and any reference, auxiliary, or termination.
- Estimate power consumption. Knowing the operating frequencies of the FPGA core and combining this with an estimate of the FPGA logic and I/O utilization, a first approximation of the power delivery and cooling systems can be determined. Refer to the **FPGA Thermal Management** section for more details.

Note: This is a good time to look at already existing IP for the FPGA core and I/O interfaces (in-house or on the FPGA vendor website) .

FPGA Schematic Symbol Generation

Describing the connectivity between components on the board, so that the actual layout of the board can be performed later, is an important step in the design flow. Since FPGAs are customizable, a design-specific symbol for the FPGA should be generated. This symbol defines how internal signals are connected to the FPGA package. On the board side of the design, this symbol serves to connect board signals to the FPGA package.

Early discussions between FPGA designers and board layout engineers are often a good way to reduce the number of design implementation iterations (cases where either the FPGA or the board layout implementation succeeded but not both). Together they can define rules that are efficient from the perspective of both sides.

The process of creating an FPGA symbol starts with the special-purpose pin assignments, for example, power, ground reference voltage, configuration pins, etc. It is important to specify the connectivity for these pins for correct board layout routing, whether or not displaying such pins within the design schematic capture environment often varies with designer preferences. The next step links the design logical I/O with the physical pinout of the targeted package. This can be done within the FPGA environment, then exported to the board layout environment or directly entered within the later environment. As Figure 3 shows, in the FPGA environment, the user can graphically specify pin assignment, then during implementation, the place and route tool automatically assigns any unassigned pins. The FPGA environment adds the benefit of performing Design Rule Checks (DRCs) that ensure legal pin assignment.

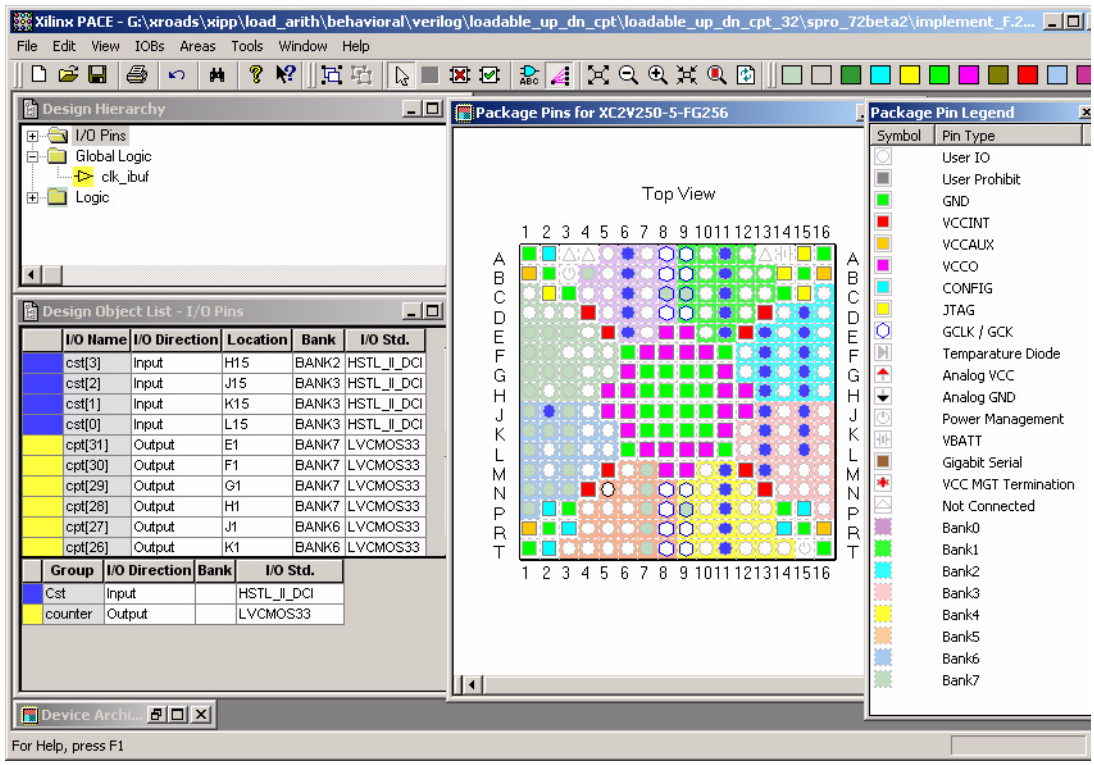

Figure 3: **Pin Assignment in Xilinx PACE™ Tool**

The last optional step is to create fractured symbols for the FPGA. Managing, displaying, and documenting symbols with hundreds of pins is not an easy task. Hence, the ability to partition symbols into multiple sub-symbols is handy. It allows designers to define manageable sub-symbols based on their preferences or company guidelines. Figure 4 shows the Mentor Graphics® FPGA Boardlink™ wizard which guides the designer through the process of FPGA symbol generation. Once the design-

specific symbol is generated, it is then used in the schematic capture environment to describe the board connectivity.

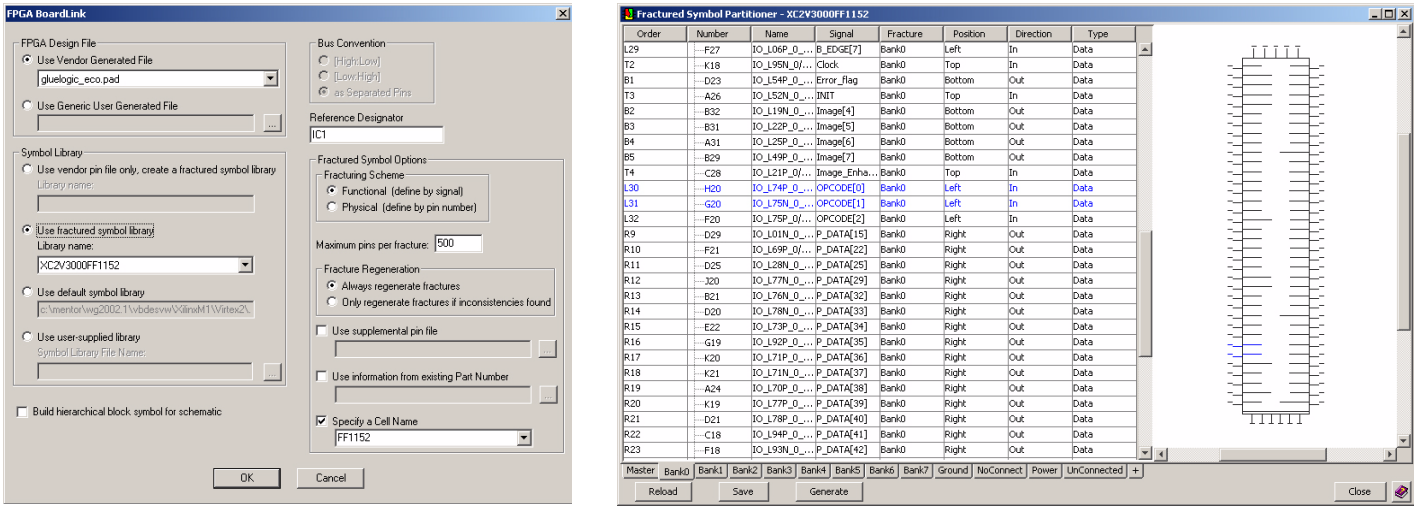

Figure 4: **Mentor Graphics FPGA Boardlink Symbol Creation Wizard**

PCB Stackup Definition

Now that the board active components are defined, it is possible to look at how the board itself should be constructed. Here are some of the specific design parameters that help to define the board stackup, material, and geometries:

- Package characteristics and I/O number and spacing are factors to determine the number of signal routing layers required for the board. Consider the board's overall cost. For instance, choosing a slightly larger package can result in a cheaper board because it reduces board manufacturing constraints.
- The data and edge rates of inter-chip links constrain the type of material, geometries, routing possibilities, and acceptable manufacturer tolerances.
- Routing density are certainly to be evaluated when defining the PCB stackup.
- A list of the different power and reference voltages required for the board components are necessary to get an idea of the number and layout of the board reference planes to be created.
- FPGA breakout techniques such as blind, buried, or micro vias processes, for instance, can help escape large BGA packages and lead to cost savings by reducing the signal routing layer count.
- Product budget, board target price, and manufacturing requirements impose constraints on material properties, via sizing, etc.

A good source of information is to look at package information and manufacturing requirement sections of the FPGA vendor's databook. In addition, it is also very valuable to allocate time to discuss the above items with the PCB manufacturer.

FPGA Thermal Management

With data rates supported and density of resources now available within FPGAs, some attention should be brought to FPGA thermal management. Estimating FPGA power consumption serves to determine whether a cooling system is required, and if it is, what the best cooling strategy is for that particular application. The ultimate goal is to ensure FPGA longevity and to account for the 3D space requirements around the FPGA, in case an active or passive cooling system is needed.

- If the design logic or I/Os have not yet been created, the FPGA vendor provides application notes and documentation like interactive spreadsheets (Figure 5) to guide the designer through the FPGA power estimation process. This process involves estimating design resources, clock domains, toggling rates, I/O standards, and many other factors that make the FPGA power consumption very much design dependent. The more data the user enters into the interactive spreadsheet, the more precise the FPGA power consumption estimation will be. See **http://www.xilinx.com/power**. Furthermore, at the FPGA synthesis level, while describing the device behavior, there are many techniques that help in managing FPGA power. This is illustrated in this Xcell article from Synplicity®, Inc., at: **http://www.xilinx.com/publications/xcellonline/partners/xc_synplicity44.htm**.

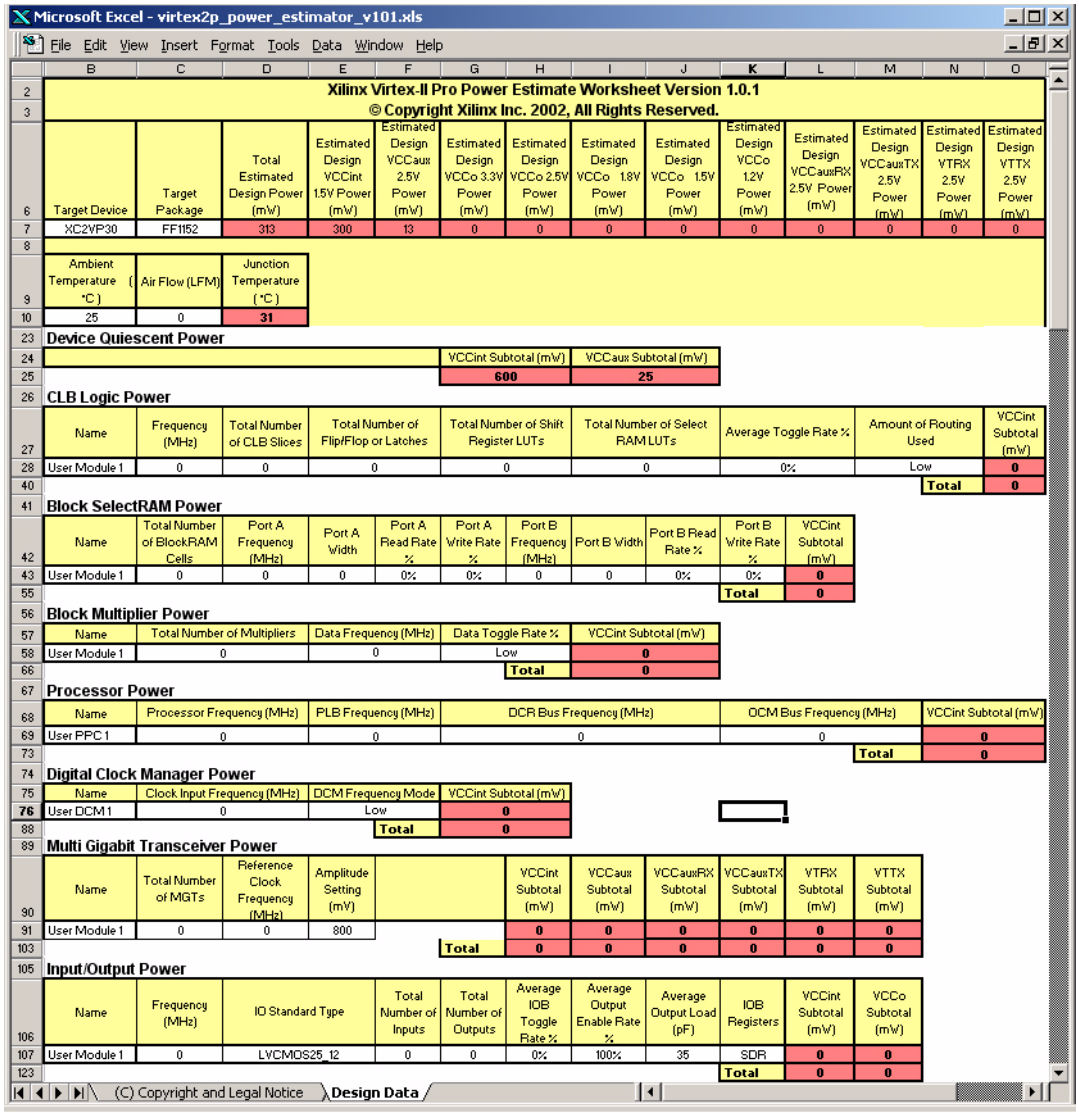

Figure 5: **Pre-Implementation Power Estimation Spreadsheets for the Different Xilinx Device Families**

- If the design logic and I/O have been specified and the design runs through the FPGA development environment (it does not have to be a placed or routed), there is an option within the development software to report the power estimation for the design.

As Figure 6 illustrates, the Xilinx Xpower tool calculates power dissipation for the entire device but also allows the user to perform individual nets analysis.

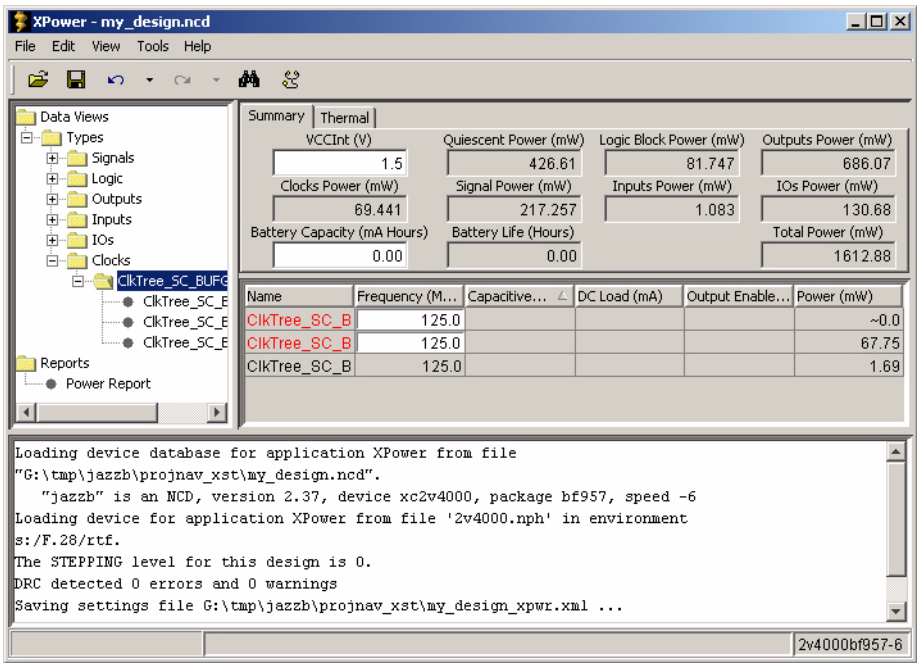

Figure 6: **Post-Implementation Power Dissipation Analysis in XPower**

Static Timing Analysis

Static Timing Analysis techniques were originally employed for gate-level event-driven simulations to verify both functionality and timing of ASIC internal logic. Because of the growth in gate count availability, previous methodologies required an ever longer time to generate, simulate, and analyze results with no guarantee of exhaustive verification. For the same reasons, these analysis techniques have become valuable for functional verification of FPGA internal logic as well.

Similarly, the exponential growth of components and component integration at the board level has led to a greater number of links between chip I/Os, making it difficult to generate test vectors that extensively verify the board's timing. Thus, to perform exhaustive coverage of synchronous and asynchronous paths between components on a PCB, the same methodology (and sometimes tools) can be used. This helps derive layout placement and routing constraints which ensure that setup and hold times are respected. Bus asynchronous transactions and clock tree analysis can also be performed.

Pre-Layout Timing Analysis

The FPGA implementation tool set has an option to generate design specific I/O timing models that can be plugged directly into timing analyzer tools like Mentor Graphics Tau™. For instance, Figure 7 shows an FPGA with an external memory interface setup and hold time analysis. These analyses have the benefit of being exhaustive, fast, and do not require any kind of test vectors as opposed to timing simulations.

Performing analysis at the beginning of the design cycle when errors can be quickly corrected is always attractive when compared to the significant efforts involved in fixing errors found after board layout or prototyping stages.

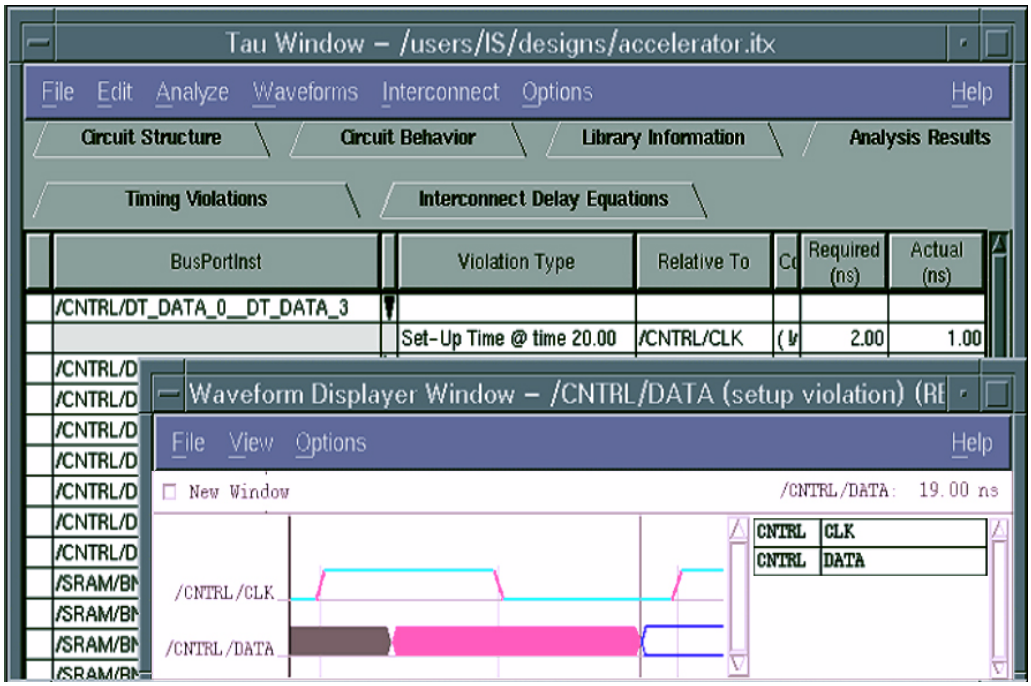

Figure 7: **Setup and Hold Time Static Timing Analysis Example Using Mentor Tau Tool**

Post-Layout Timing Analysis

Performing post-layout timing analysis helps to verify chip-to-chip path timing. As for pre-layout timing analysis, simulations are exhaustive, fast, and do not require the creation of stimulus. In addition, trace delay is calculated directly from the board layout tool database. From such simulations, the user is able to verify timing margins as close to the implemented board as possible.

Signal Integrity (SI) Analysis

During this step of the overall PCB design flow, thorough analyses are made to determine the solution space in which different communication channels (between the FPGA and its environment) can work. Performing signal integrity analysis within time and frequency domain explores proposed link topology (PCB construction, driver, receiver, connectors, vias, etc.). The goal is to evaluate and minimize distortions affecting signals traveling down the path from driver-to-receiver through the PCB, thus guaranteeing signal fidelity. By analyzing effects such as impedance discontinuities, crosstalk effects, and electromagnetic radiation, it is possible to generate realistic constraints that will effectively guide the PCB place and route tool. The first step in this process is to examine FPGA I/O needs and constraints from the components interfacing with the FPGA. From there, approximate I/O standard and termination schemes are derived. The third step, discussed later in detail, is to refine the I/O parameters through simulations. The fourth step is to measure signal quality on the board prototype and perform SI optimizations if necessary.

Pre-Layout SI Analysis

Nets with the fastest edge rate or the most critical timing, the greatest number of loads, or the longest nets are often the most subject to SI degradations and, therefore, are examined first. Once critical nets are identified, the next step is to enter for each of them the topology from driver(s) to receiver(s) into a simulation environment. Depending on the edge rate or the simulation accuracy desired for the analyzed link, simulations can be performed using behavioral models (accurate/fast) or transistor-level models (most accurate/slow). The purpose of setting up these simulations is to perform all kinds of "what-if" analyses to help define the physical and electrical parameters of the considered link. Figure 8 shows a topology example along with simulation results.

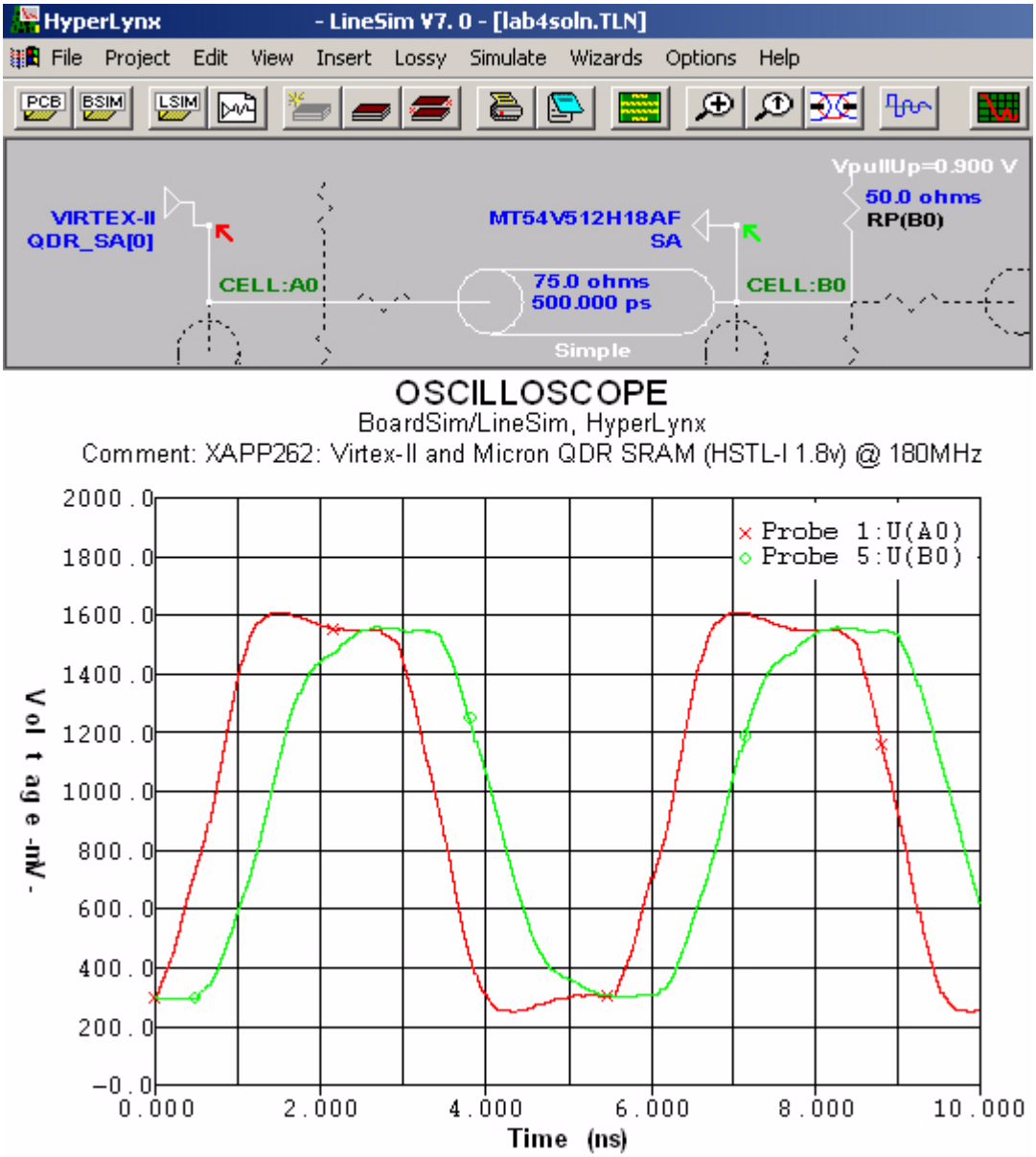

Figure 8: **Pre-Layout SI Analysis Using Mentor's HyperLynx™ Suite**

Performing such simulations helps to:

- Define the maximum link length and other net properties that meet noise margin criteria.
- Define FPGA driver buffer properties, such as the I/O standard, driver strength, and slew rate that suit the envisioned link, while minimizing SI, EMI, or power consumption phenomena.
- Similarly define FPGA receiver buffer properties.
- Perform crosstalk analysis to ensure that adjacent nets do not affect each other's signal quality.
- Define termination strategies. Evaluate external or internal termination schemes that will minimize ringing and other phenomena adversely affecting signal propagation quality. For more information on digitally controlled impedance and on-chip termination technology visit the XCITE technology portal at: http://www.xilinx.com/xlnx/xil_prodcat_product.jsp?title=v2_xcite_page.

  Some termination schemes are serial or parallel, with AC or DC coupling. Each of these schemes has pros and cons in terms of performance, power, and number of external components on the board. Evaluating these schemes is about making trade-offs that make the most sense in the designer's particular system environment.

Exploring signal integrity effects in the time and frequency domain early in design process (before any hardware prototype is actually built) can seem tedious or ineffective at first, but always proves valuable in terms of lowered cost and improved time to market. For example, this flow eliminates the time and cost involved in building another prototype, accounting for components that were lost from the previous prototype board(s), and the cost and time of debugging what, in fact, could be regarded as design flaws.

Post-Layout SI Analysis

After the board place-and-route phase is completed, performing SI analysis on the most critical nets of the design helps ensure that the noise-level in the implemented link is within the margins defined by the design specifications. The analysis process is often similar to that of the pre-layout step with a particular emphasis on crosstalk and EMI analysis, since at this point in the design flow, the actual layout of traces is known and can be simulated. Should there be any net found failing the requirements, then further "what-if" analyses could investigate configurations that would satisfy the design requirements. Subsequently, routing for these paths would be redone. Figure 9 shows a routed pair of nets extracted for SI analysis, so that post-layout analyses (eye diagram on the left) are performed and then layout constraints (spreadsheet on the right) are eventually adjusted if design specifications are not met.

Power Distribution System (PDS) Analysis

FPGA designers are faced with a unique task when it comes to designing the PDS. Most other large, dense ICs (such as large microprocessors) come with very specific bypass capacitor requirements. Since these devices are only designed to implement specific tasks in their hard silicon, their power supply demands are fixed and only fluctuate within a defined range. FPGAs do not share this property. An FPGA can implement a practically infinite number of applications at undetermined frequencies and in multiple clock domains, so it is too complex to characterize device transient current demand in the FPGA databook. This is why Xilinx provides additional documentation and application notes, such as XAPP623

(http://www.xilinx.com/xapp/xapp623.pdf), which guide the user through the process of designing a robust PDS for a particular application.

The purpose of PDS analysis is to evaluate transient current demands by digital devices and to provide paths that allow these currents to flow with a minimum of constraints. Inductances in the current paths are a source of supply voltage degradations or distortions (e.g., ground bounce and power supply noise). One possible effect is IC signals not switching when they are supposed to (incorrect data, timing distortion). Another more common effect is increased system jitter, which degrades timing and can also lead to failures. In both cases, this results in a system that does not operate properly or operates out of manufacturer's specifications.

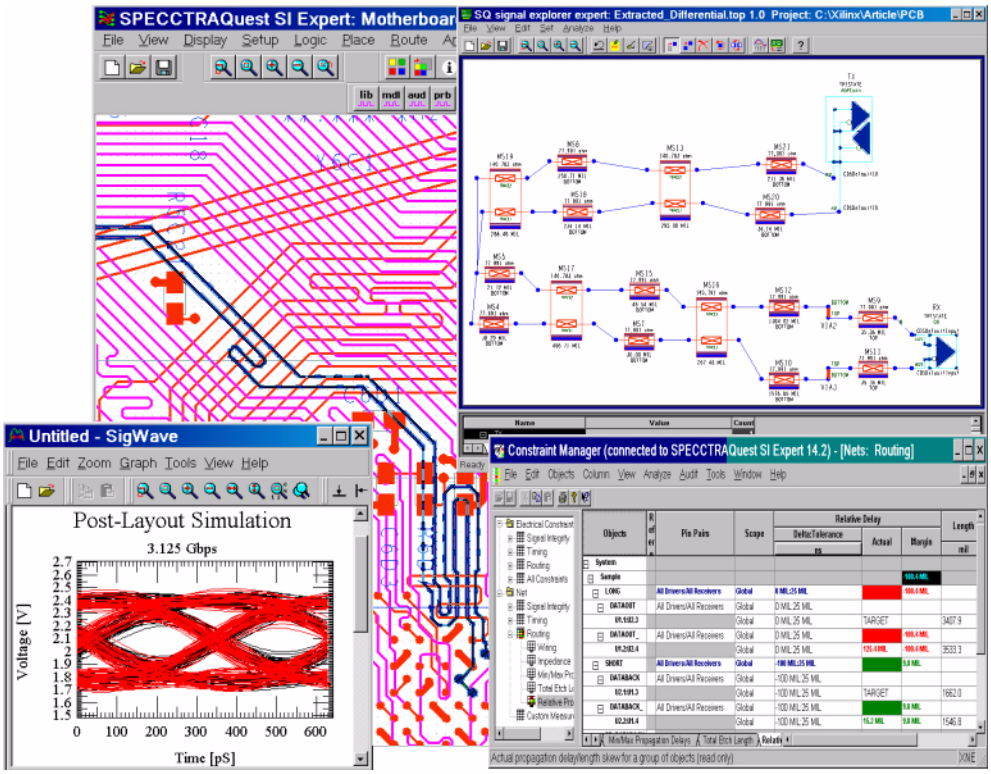

Figure 9: **Post-Layout SI Analysis Example Using Cadence® SPECCTRAQuest™**

The first step in this process is to examine the utilization of the FPGA for its static and transient current requirements. Transient requirements are defined by the number and frequency of the design's clock domains, DCM usage, amount of switching logic, simultaneous switching outputs, etc. Next, an approximate decoupling network should be designed to fit these requirements. The third step is to refine the capacitor quantities and values through simulations. In the fourth step the full design is built, and in the fifth step it is measured. Measurements consist of oscilloscope and possibly spectrum analyzer readings of power supply noise. Depending on the measured results, further iterations through the part selection and simulation steps could be necessary to optimize the PDS for the specific application. These analyses have implications on the board layout, because not only the number and value of the decoupling capacitor network but also the capacitor placement are important. Figure 10 shows (for a specific design) a power plane impedance simulation before and after adjusting the power delivery system. By carefully selecting the decoupling capacitor network, the plane impedance across the FPGA operating frequencies can be

lowered. This means transient currents resulting from the device demand are accommodated more quickly and thus the power supply noise is reduced.

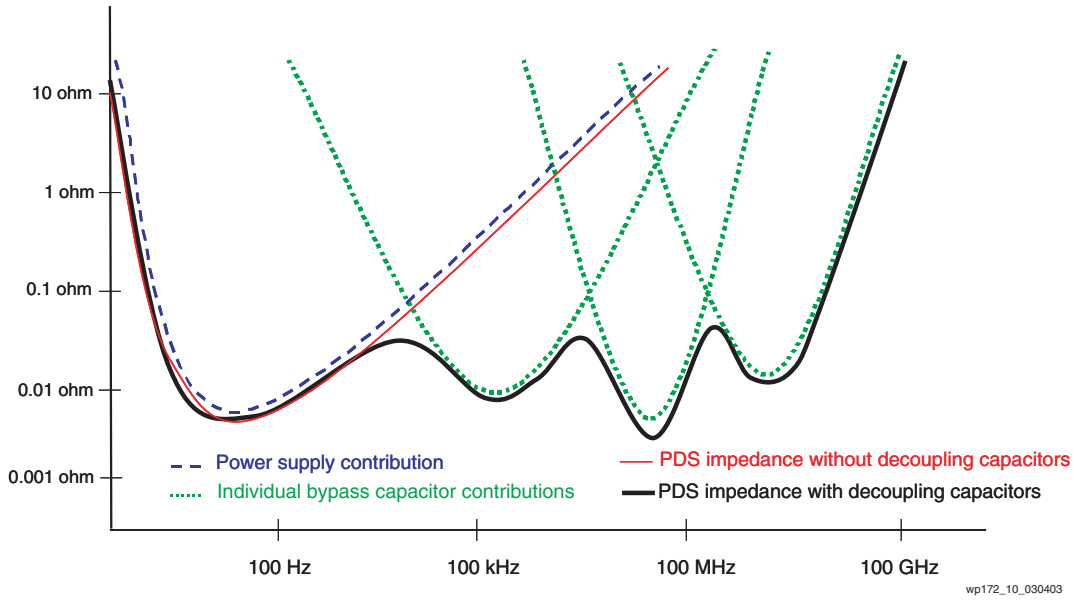

Figure 10: **Perform PDS Analyses To Define the Decoupling Network That Supplies the FPGA Across the Design Operating Frequencies**

Design for Test

As board density increases, it becomes more and more difficult to physically probe PCB signals. Routed signals might not be accessible because they are buried in the dielectric, and chip I/Os might not be accessible because of dense IC packages like flip-chip or ball-grid array. Furthermore, for high-speed designs, probes themselves can sometimes affect the signal quality or measurement. The designer can leverage the use of FPGAs to ensure board testability. Exploiting the reprogrammability of the devices allows the creation of a built-in self-test logic (BIST) that exists only during testing. As a result, the test has no area overhead or performance penalties to the normal system operation

Designers can create a test design for each link between the FPGA and its environment which will generate output stimulus and monitor input signal activity. In this way, designers can use the available FPGA logic resources to compare acquired inputs with expected data, then identify possible discrepancies and send results to the JTAG debug port or any display device (if there is one in the application).

If it is possible to insert probes at the end of the considered link, then signal degradation through the communication channel can be evaluated for specific stimulus by using the FPGA as a signal pattern generator. Using FPGAs as test controllers is valuable because it makes board testing and verification more partitionable, as well as easier and faster by providing more flexibility in performing PCB testing and debugging. .

Conclusion

Throughout this document, FPGA-related analyses of the overall PCB design have been presented. Methodologies performing for a large variety of analyses and simulations early in the design flow before any PCB construction even begins are recommended. The objective is to build knowledge of constraints and requirements from the FPGA on its PCB environment and vice versa. Results from such analyses

represent the solution space in which the design should exceed PCB and FPGA requirements. Then, by converting the analyses' results into constraints for the board layout place and route tool, the chances for first time PCB design success will be maximized. For a particular design, it is ultimately up to the developers to assess which analyses should be performed and the degree of thoroughness for each analysis. The potential benefits of this methodology depends also on the complexity of the PCB and FPGA design, past experience of the developers, and other factors. The ultimate goal of this process is to minimize the product's time-to-market and development cost.

References

The following references are provided:

Table 1: PCB Design Products Provided by Xilinx Partners

Company	Product	Functionality	URL
Cadence® Design Systems	Concept® HDL (includes Build Physical Wizard)	Schematic capture and FPGA symbol generation	http://www.cadencepcb.com/products/concepthdl
	OrCAD Capture®	Schematic capture and FPGA symbol generation	http://www.cadencepcb.com/products/capture
	SPECCTRAQuest™	High-speed PCB design and analysis	http://www.cadencepcb.com/products/specctraquest
	Allegro®	PCB layout	http://www.cadencepcb.com/products/allegro
	OrCAD Layout®	PCB layout	http://www.cadencepcb.com/products/layout
Mentor Graphics®	Design Architect™	Schematic capture	http://www.mentor.com/designarchitect
	Board Architect™	Schematic capture	http://www.mentor.com/boardarchitect
	FPGA Boardlink™	FPGA symbol generation	http://www.mentor.com/fpga_boardlink
	ICX™	High-speed timing and signal integrity analysis	http://www.mentor.com/icx
	TAU™	High-speed timing analysis	http://www.mentor.com/tau
Product Acceleration, Inc.	DesignF/X™	FPGA placement and pin assignment optimization	http://www.prodacc.com/html/webpages/Products.htm#Product_DesignF/X
Synopsys®	HSPICE®	High-speed signal integrity simulation tool	http://www.synopsys.com/products/avmrg/hspice_ds.html

Other Resources

For more information, refer to the following resources:

- Xilinx portal for all design considerations (PCB design, signal integrity, thermal design, and power supply considerations) at:
  http://www.xilinx.com/xlnx/xil_prodcat_landingpage.jsp?title=Signal+Integrity

- *Digital System Engineering* by William J. Dally, John W. Poulton. This is a reference for electrical design of digital systems. It discusses power distribution, noise management, signaling, timing, and synchronization.
- *High-Speed Digital System Design: A Handbook of interconnect Theory and Design Practices* by Stephen H. Hall, Garrett W. Hall, and James A McCall. This is a reference for high-speed interconnect effects in digital designs.

Revision History

The following table shows the revision history for this document.

Date	Version	Revision
03/13/03	1.0	Initial Xilinx release.