

A Fundamentally Different Approach to FPGA I/O Design

Agilent used I/O Designer to achieve radical reductions in design cycle time.

by Bruce Riggins
Product Marketing Manager
Mentor Graphics
bruce_riggins@mentor.com

New problems require new solutions; this has never been truer in the realm of FPGA design than it is today. Devices with higher and higher pin counts have placed a significant strain on classic, over-the-wall methodologies. These techniques, which worked for many years, typically addressed FPGA and PCB design efforts as two distinct disciplines, where the FPGA designer defined the I/O assignments of the FPGA and passed those assignments to the PCB designer.

Ignoring for a moment the ramifications on the PCB, this process generally worked pretty well, especially with smaller devices and simple system performance requirements. But as FPGAs have grown larger and larger, nearing (and soon exceeding) 2,000 pins, these design philosophies simply fall apart. These devices demand a fundamentally different approach to FPGA I/O design - an approach that considers, concurrently, the effects that I/O assignments have on the FPGA and PCB.

In this article, I'll chronicle a successful deployment of this new process, based on a real-world design completed by Agilent Technologies using Mentor Graphics's I/O Designer.

Design Overview

Before discussing some of the reasons for Agilent's success, let's review the characteristics of their design:

- Eleven 1,148-pin Xilinx® Virtex™-4 LX series FPGAs
- Large DRAM buses on several devices (up to 250 pins in some cases)
- Multiple high-data-rate (10 Gbps) inter-FPGA buses approximately 48 pins wide
- Wide physical interfaces to PHY devices and backplane
- Very few spare pins
- Complex FPGA restrictions
- Fourteen differently-sized banks in the Virtex-4 device
- Clocks associated with I/O must be on specific signals
- I/O restrictions – for example, clock-capable inputs do not support outputting LVDS
- Mixed voltage environment (1.8V/2.5V/3V/3.3V) for different banks
- A 26-layer PCB
- Eight inner tracking (signal) layers
- 9,800 components
- 9,500 nets

By any measure, this was a very complex project.

From previous experience, Agilent estimated that they would need four to eight weeks per FPGA just for the I/O pin assignment process. In addition, I/O restrictions hinted that problems would be likely (and indeed, subsequent designs confirmed that). Agilent also knew that it was difficult to maintain links between the FPGA and board designs as the pinouts changed. Given these challenges, Agilent felt compelled to consider new design strategies.

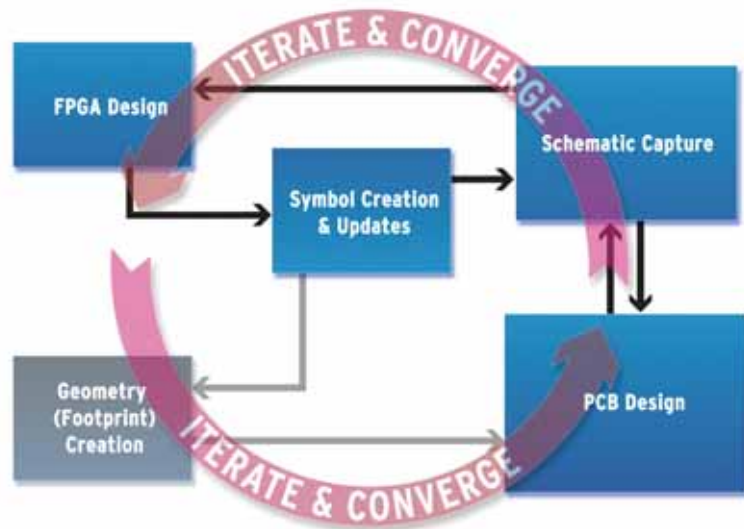


Figure 1 – A typical FPGA-based board-level design flow

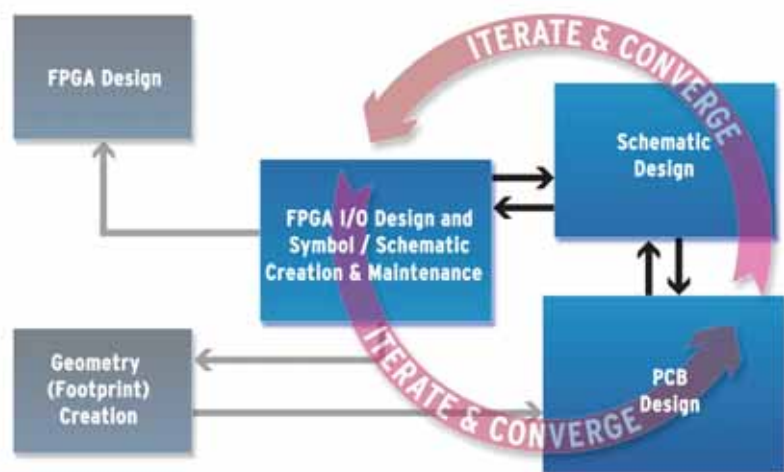


Figure 2 – A typical FPGA-based board-level design flow using I/O Designer

Simplifying the FPGA/PCB Design Process

Figures 1 and 2 illustrate typical, FPGA-based board-level design flows. On the surface these two figures look the same. But closer inspection reveals that the use of an FPGA/PCB co-design tool such as I/O Designer can simplify the flow, reduce error-prone tasks, and ultimately decrease design cycle times.

In a classic process (Figure 1), the FPGA designer works to develop an I/O assignment that is eventually deemed worthy of passing to the rest of the team. Before any

downstream processes can proceed, the designer must create a schematic symbol (or symbols) and PCB footprint for the FPGA. The symbol and footprint are then used by the schematic and PCB designers in their portion of the design effort.

However, if either of these individuals require changes to the FPGA's pin assignment, they typically go back to the FPGA designer, who makes the changes in the FPGA tools and then feeds those changes to the librarian, who updates the schematic symbols. (Component footprints usually

change very little, if at all, since the geometries are mechanical in nature and are not affected by new pin assignments.) These are then given to the schematic designer and the project proceeds.

Simply speaking, converging on a completed design requires repeated iteration of these steps. This process is highly prone to mistakes because of the large amount of manual effort needed to keep the FPGA, symbols, schematics, and PCB layout synchronized.

Figure 2 also demonstrates a typical board-level design flow, but with I/O Designer introduced into the middle of the process. This is comparable to the flow adopted by Agilent, where I/O Designer was used for FPGA I/O pin assignments and for all FPGA-related symbol and schematic creation and maintenance. This allowed Agilent's engineers to focus their skills where they had the most value: the creation of intellectual property. They were thus able to iterate and converge on a smaller, more manageable subset of the overall system. (The engineers continued to use Xilinx FPGA tools at various points in the process for traditional place and route and synthesis and to check for compliance using Xilinx design rule checks.)

I/O Designer seeks to bring FPGA and PCB designers together in a common environment so that each team member can see the ramifications of FPGA pin assignments on the entire system. The mechanism that I/O Designer uses to accomplish this is conceptually simple: a view of the PCB with a dynamically assignable FPGA library element instantiated in place of the FPGA's typical PCB footprint.

The FPGA component, having come from a library of "intelligent" FPGA devices, assists in making proper pin assignments (checking for differential pairs, I/O banking rules, and SSO, for example). Because changes to these assignments are shown in real time, the potential effects on the rest of the board are immediately obvious.

Figure 3 is a screenshot of a section of Agilent's board in their PCB tool. Agilent used I/O Designer to optimize the ratsnest (some of which are shown in green and orange) across the entire board.

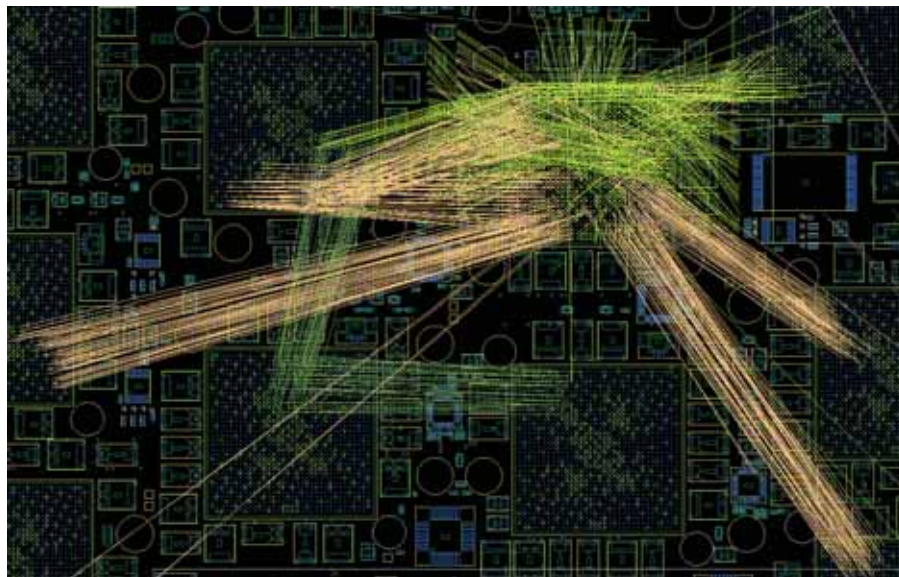


Figure 3 – A section of Agilent's 11-FPGA PCB

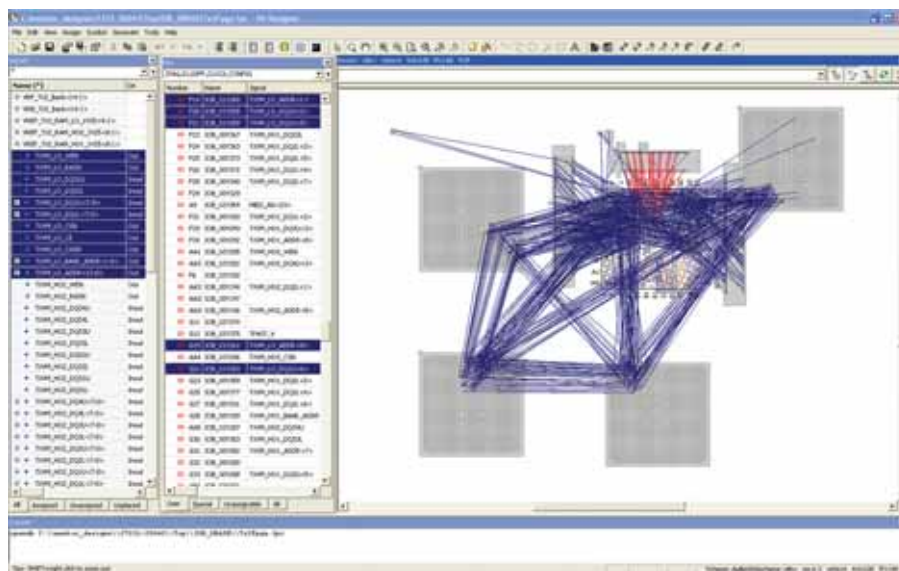


Figure 4 – A section of Agilent's design within I/O Designer

To demonstrate how I/O Designer assisted in this effort, Figure 4 shows the same Agilent design within I/O Designer after pin optimization. The area on the right is the "multi-component" window, which is a view of the PCB, and the FPGA currently undergoing active assignment. Other windows convey signal, pin, symbol (not shown), and console (tran-

scripting) information. By displaying the inter-component connections from a board-level perspective, I/O Designer allowed Agilent to optimize the pin assignments for each FPGA.

Connecting to the FPGA Tools

I/O Designer is not an FPGA design tool: it can neither route nor synthesize. As such,

it can and must read and write the files necessary to close the loop with the FPGA designer (making pin-assignment changes in the schematic or PCB tools, or within I/O Designer, is of very little use if those changes cannot be communicated to the FPGA tools). To assist with this closure, I/O Designer can generate or update the associated place and route and synthesis constraints files through a simple pull-down menu.

The Overwhelming Benefits of Hierarchy

Flattened design techniques implement the project as one large collection of sheet after schematic sheet. In this approach, engineers begin by placing components on a schematic sheet. When that sheet is filled, they move to sheet two, then sheet three, and so on until the design is complete. This results in a database that conveys very little useful information about the logical architecture of the system. Without regular involvement in the project, it is nearly impossible to comprehend the design by looking at the schematics.

Hierarchical approaches, on the other hand, start with a top-level view of the system, then devolve the system into smaller and smaller more manageable subsystems. At some point, this devolution ends at the board or chip-level and the design begins.

Using hierarchy, every level of the system can be understood by anyone with even a cursory understanding of the project. As a result, the design database not only defines the physical realization of a subsystem, it documents the purpose of the subsystem in a way that a linear collection of schematic sheets simply cannot. Another benefit of hierarchy is that it forces engineers to consider the architecture of the design, not just its implementation.

“We realized early on that hierarchy would enable us to more fully exploit the features in I/O Designer,” said Ross MacIsaac, a senior design engineer at Agilent’s South Queensferry site in Scotland. “As such, although hierarchy was already used extensively within Agilent, we were careful to architect this system to ensure that I/O Designer, as

well as our established EDA tools, would deliver us the most productivity. There is no doubt that the use of hierarchy reduced our design cycle time.”

Why the focus on hierarchy? Because I/O Designer, when used in tight combination with a hierarchical design process, can provide significant productivity improvements. A structured, hierarchical approach enables I/O Designer to easily generate and – as the design matures, maintain – the symbols and schematics that make up the FPGA-related portion of the design. Thus, the conceptually simple burden of keeping the schematic view of the FPGA synchronized with the rest of the database, while eliminating manual, tedious, error-prone tasks, is considerably reduced or eliminated.

Caveats and Lessons Learned

Much of what Agilent undertook with this effort had never been attempted before, at least on this scale. Along the way, they learned a few lessons, as did Mentor. Here are some of their recommendations:

- Use as much of I/O Designer’s automatic symbol and schematic creation and maintenance features as the process will allow.
- Finely partition the FPGA. Use an individual symbol for each signal bank, power block, configuration block, and JTAG block. This greatly reduces the potential for errors.
- Let I/O Designer handle the creation and maintenance of the schematic sheets that contain the FPGA logic banks. Do not mix power, configuration, and JTAG blocks with logic blocks, and partition the design in such a way as to separate the I/O Designer-maintained schematics from user-maintained schematics.
- Any team-based design approach, especially one with multiple FPGAs, requires that the project be forked and merged at several strategic points in the design cycle. Although I/O Designer can certainly help in these situations,

getting the most from it in these cases requires careful planning and forethought. In other words, do not expect the tool to magically solve the project management problems. Take the time to learn how it can help, then deploy it accordingly.

- Cleverly partitioning the design into sections that can be attacked as homogenous blocks, and an application of hierarchy that enables the design to be quickly reconstructed from those blocks, can significantly improve design team efficiency.

Conclusion

New tools and approaches can significantly reduce project cycle times while simultaneously producing superior results. By cleverly architecting their design and molding their processes to take advantage of their EDA tools, Agilent was able to realize substantial productivity gains. Specifically, they were able to reduce their pin assignment effort from four to eight weeks per FPGA to one to two weeks per FPGA (including the time needed to compile the FPGA to check design rules), going from schematic start to PCB layout complete in less than 10 months while reducing overall system design effort by roughly 50%.

“We were aware of the emergence of I/O Designer,” said MacIsaac. “We opted to introduce it into our process at a point in the design cycle that was later than we were comfortable with, as there was a strong preference to use proven tools.

“However,” he continued, “the size of the task merited taking that risk. The effort involved in other schemes was just so much that it was expected to exceed any tool teething problems. Management was prepared to take the risk of introducing this new technology. This board was the right problem at the right time for the tool [I/O Designer] and the FPGAs. Both gave us the ability to do the design and alter it, quickly and reliably.”

For more information about Mentor Graphics’s I/O Designer, visit www.mentor.com/products/pcb/expedition/system_design/io_designer. 