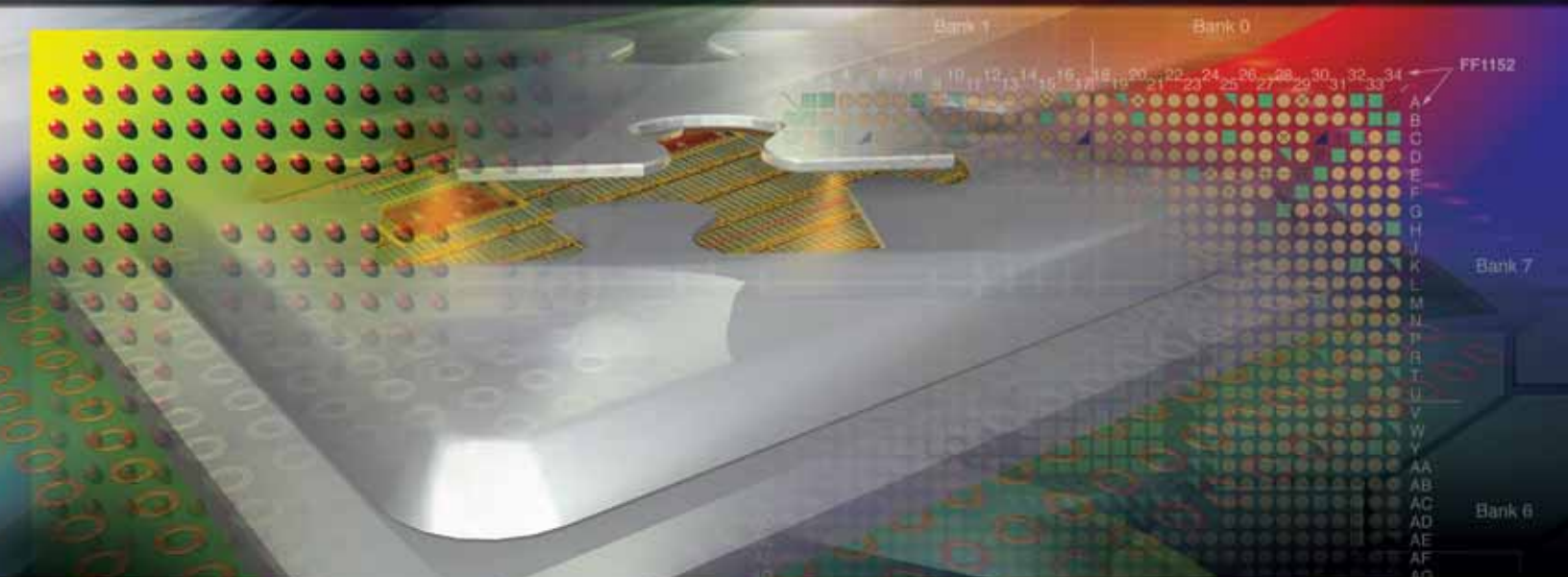


Solving FPGA I/O Pin Assignment Challenges

Here's a step-by-step methodology to help you pinout complex FPGAs.



by Brian Jackson
Product Marketing Manager
Xilinx, Inc.
brian.jackson@xilinx.com

I/O pin assignment is one of the main challenges facing designers integrating large FPGA devices onto PCBs. Many designers find the process of defining the I/O pin configuration, or “pinout,” of large FPGA devices and their advanced BGA packages an increasingly difficult task for a seemingly ever-expanding number of reasons. But with a mix of smart I/O planning and new tools, you can remove the pain from the pinout process.

The task of defining an I/O pinout from FPGA to PCB is a major design chal-

lenge that can make or break a design. You must balance requirements from both the FPGA and PCB perspectives while designing both sides in parallel. If you prematurely optimize a pinout specifically for the PCB or the FPGA, it can lead to design issues in the other domain.

You need to be able to visualize both the PCB placement and FPGA physical device pins, along with the internal FPGA I/O pads and related resources, to understand the ramifications of your pin assignment choices. Unfortunately, as of today there isn't just one tool or methodology to address all of these co-design concerns.

What you can do, however, is combine various techniques and strategies to optimize the pin-planning process and add

new co-design tools like Xilinx® PinAhead technology to devise an effective pinout methodology. Xilinx includes PinAhead in the ISE™ software design suite 10.1.

At Xilinx, we have developed a rule-driven methodology in which we define an initial pinout that considers both the PCB and FPGA requirements, allowing each design group to begin their respective design processes as early as possible by using a pinout that should be very close to the finalized version. If the design requires changes because of PCB routability or internal FPGA performance issues late in the process, this methodology is such that those issues are typically localized, requiring you to make only small changes in either design domain.

Step 1: Evaluating the Design Parameters

So where should you start? You should begin to formulate an I/O strategy as early as possible. But you may find this task difficult in the absence of an optimal tool for the job or a complete netlist.

First, let's examine the PCB physical parameters and limitations by answering a few questions:

- What is the preferred layer count, trace width, and via size?
- Do the PCB parameters limit the FPGA package types that you can use, such as BGA?
- Are there any fixed interface locations for the FPGA on the PCB? Other chips, connectors, or placement restrictions?
- Which high-speed interfaces need special attention?
- Can you visualize a placement strategy to enable the shortest interconnect?

You may find it helpful to draw a diagram of the proposed PCB placement – including all major components with critical interfaces and buses – to determine

the best FPGA pin assignment locations. Make sure that you draw the components on the side of the board where you plan to mount those components. Make note of interfaces that will require special attention, such as high-speed buses and differential pairs (Figure 1).

Next, examine the layout of the FPGA device to understand where the physical resources exist on the silicon. List the various voltages and clocks you are using in the design to begin to isolate the interfaces the design will require. Then determine if your design uses specific I/O interface resources such as giga transceivers (GTs), BUFRRs, IODELAY, and digital clock managers that require you to define and route I/O pins in close proximity to each other.

Now it's time to locate the FPGA resources, such as PowerPCs, DSP48s, and RAM16s, in the design. It makes sense for you to target any related I/Os to the I/O banks closest to those resources. See if you can group any of the I/O signals into interfaces; this will help during pin assignment. Finally, determine the configuration mode for the FPGA.

Step 2: Defining Pinout Requirements

Once you understand the main FPGA interfaces and have created a mockup of the physical layout, you can start to define the pinout. Some designers like to use a spreadsheet containing all of the I/O signals to keep track of the pins. You can group them by voltage, by clocks, by interface, or by bus. This method is really quite valuable because it helps you begin to formulate the groups of signals you will assign in close proximity. At this stage, you should also identify critical interfaces that must exit the device on a particular edge or use outside physical pins for optimal PCB routing.

After examining both the FPGA and PCB requirements and defining the major interface locations, the next step is to begin assigning pins to I/O banks based on all of the preceding criteria. This is where the real work begins. In the current flow, pin assignment is a time-consuming task that can involve a lot of trial and error to solve any performance and signal integrity concerns. Designers have traditionally performed this task freehand, because EDA and chip ven-

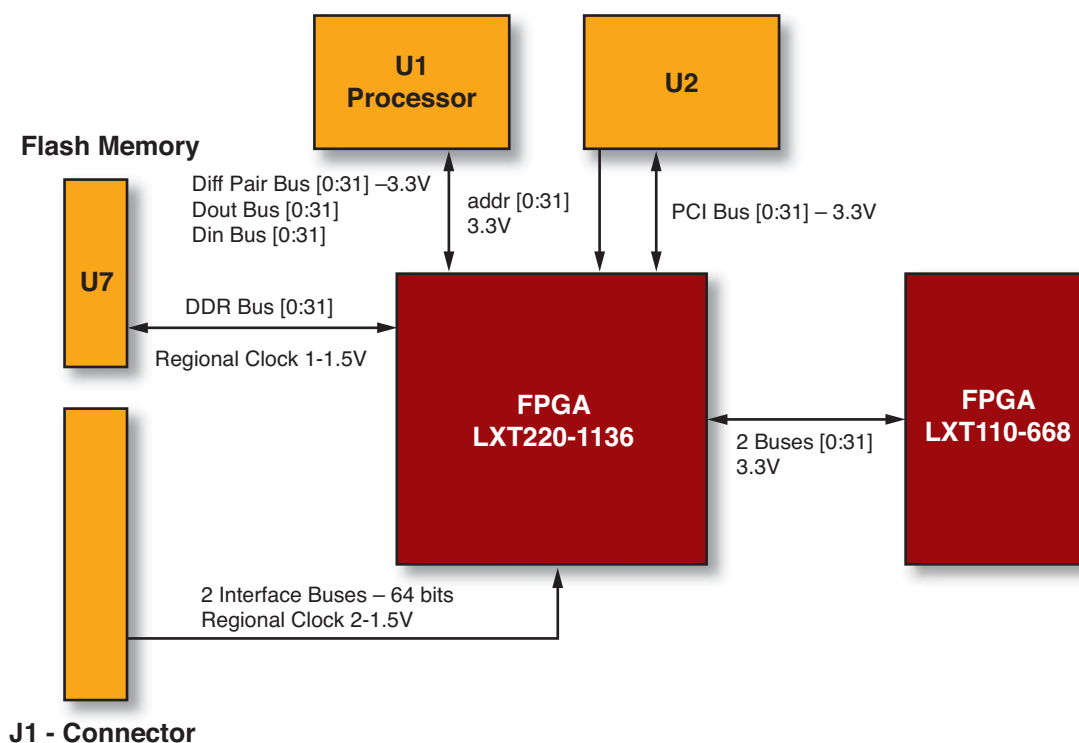


Figure 1 – Create a PCB connectivity diagram

dors didn't offer tools to effectively help designers visualize the two domains.

But now, Xilinx has a tool for the job. PlanAhead Lite, which we've included in the 10.1 release of the ISE Foundation™ software toolset, is a subset of the PlanAhead™ software design, analysis, and floorplanning tool. It includes a tool called PinAhead, which addresses both PCB and FPGA design concerns and makes I/O pinout configuration much easier for designers.

Rather than go into detail about every feature of the tool, let's see how we can use it in the context of an I/O pin assignment methodology. If you want more detailed information about PinAhead, including a video demonstration and tutorial, visit www.xilinx.com/planahead.

Step 3: Using PinAhead for Assigning the Pinout

The PinAhead environment displays a set of views applicable to the task of exploring and assigning I/O port signals to physical package pins or I/O pads on the die (Figure 2).

PinAhead has virtually the same graphical environment as PlanAhead software, clearly displaying the silicon I/O pads and related resources in the "Device" view and the physical device pins in the "Package" view. The views present both I/O port and physical pin information so that you can cross-select to explore the logical design and physical device resources.

You can use PinAhead in the absence of a design netlist to explore device resources or to start the I/O pin planning process. The "Package Pins" view lists device package specifications according to the device data sheets, so in most cases you won't have to cross-reference the device data sheet when configuring the pinout. The Package Pins view table categorizes I/O banks, allowing you to cross-select and highlight I/O banks in both the Device and Package views. The view clearly shows the relationship between the physical pin location and the I/O pad location on the die, which simplifies optimal I/O bank selection. The Package Pins' "Pin" view also displays information for each pin in the I/O bank.

You can begin creating I/O ports from scratch using the PinAhead interface or by importing them from a comma separated

value (CSV) format spreadsheet, HDL source file headers, or a synthesized netlist and UCF format constraints file. The "I/O Ports" view displays all of the I/O port signals you have currently defined in the design, and the bus folders display the grouped buses and differential pair signals.

You can sort the Package Pins and I/O Ports views in a variety of ways. You can toggle the list views to display a category-based list or a flat list, and with just a mouse click, sort the Package Pins view to display all available global clock- or regional clock-capable pins. You can also export the information to a CSV format spreadsheet to use as a starting point for pinout configuration.

PinAhead also has an interface to allow you to selectively prohibit I/O pins, groups of I/O pins, or I/O banks from having I/O ports assigned to them by PinAhead. You can select and prohibit pins in the Package Pins, Device or Package views. For example, you can sort the Package Pins view and prohibit all VREF pins.

PinAhead allows you to group related I/O ports and buses into an "interface." This grouping simplifies I/O port management and assignment tasks by allowing you to handle related I/O ports as a single entity. The interface grouping makes it much easier for you to visualize and manage all of the signals associated with a particular logic interface.

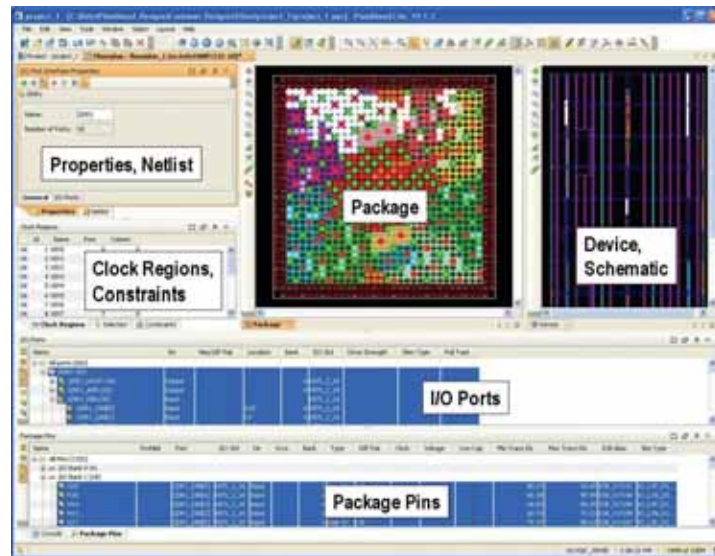


Figure 2 – The PinAhead environment

You can easily copy interfaces from design to design or use them to generate interface-specific PCB schematic symbols. The interfaces appear as expandable folders in the I/O Ports view, and you can add additional I/O ports to the interface by selecting them in that view and dragging them into the interface folder.

As you create the I/O ports, you can assign them to package pins or I/O pads. Before you do this, it pays to examine the I/O ports' initial PCB interconnect sketch and consult with the PCB designer to understand the desired locations and other considerations for placing the various I/O port interfaces. Proper bus order and edge proximity can aid PCB routing significantly and save design time.

You can assign individual pins, buses, and interfaces to I/O pins by dragging them into either the Device or Package views. You can assign the entire group of pins to the selected I/O pins using various assignment pattern modes. The available modes include "Place I/O Ports in an I/O Bank," "Place I/O Ports in Area," and "Place I/O Ports Sequentially."

Each mode offers a different assignment pattern for the I/O ports you are assigning to pins. Using these modes, you will see information about the number of ports you are placing by looking at the pop-up window on your mouse cursor's tip. This mode remains active until you have placed all of the selected I/O ports.

The Device view graphically displays all clock regions and clock-related logic objects, making I/O assignment based on clocks much easier and intuitive. Selecting a clock region will display all of the I/O banks, clock-related resources, and device resources associated with it.

PlanAhead software attempts to maintain your correct-by-assignment rules. With your guidance, the tool will assign differential pair ports into proper pin pairs. As you interactively assign I/O ports with the tool, the tool runs design rule checking (DRC) to ensure legal placement.

The tool's default setting runs in interactive DRC mode, but you can toggle it off if you choose. The tool checks for conflicting voltages, VREF pins, or I/O standards, as well as noise-sensitive pins located near GT devices. When it finds an error, the tool displays a Tooltip indicating why you can't assign an I/O port to a specific pin.

You can activate PinAhead's "Autoplace" command to direct the tool to automatically place all or any selected I/O ports to package pins. The Autoplace command will obey all I/O standard and differential pair rules and will place global clock pins appropriately. The command will also attempt to group interfaces as much as possible.

The Device view graphically displays all clock regions and clock-related logic objects, making I/O assignment based on clocks much easier and intuitive. Selecting a clock region will display all of the I/O banks, clock-related resources, and device resources associated with it. This makes the process of regional clock planning much easier by allowing exploration of the available resources along with their physical relationship.

You can also use PinAhead to place other I/O related logic in your design, such as BUFs, BUFRs, IODELAYS, IDELAYCTRLs, and DCMs. You can easily locate these objects and placement sites in PlanAhead by using the "Find" command. To selectively explore and expand logic and logic connectivity, use the tool's "Schematic" view.

You can also lock the placement of specific I/O-related logic with PlanAhead software by selecting the logic in any view and dragging it onto sites in the Device view.

PlanAhead software will only allow you to place logic on appropriate sites. A dynamic cursor identifies the proper site locations as you are dragging a logic object over the design.

Step 4: Running DRC and WASSO for Legal Sign-Off

Once you've completed your pin assignment, you can use PinAhead's vast set of DRC rules to run a sign-off DRC and ensure that the design is error-free before running the PlanAhead software implementation tools. The tool has many I/O and clock-related rules to ensure that I/O placement is legal. You can select rules using the PlanAhead software DRC dialog.

If the tool finds rule violations, it displays a DRC results table with the errors. You can select the errors to investigate them further.

PlanAhead software also includes weighted average simultaneous switching output (WASSO) analysis to help you identify potential signal integrity issues with the pinout configurations. You feed the tool the parasitic characteristics of your PCB design; PlanAhead software will analyze the various I/O banks and their neighbors and report back the utilization and status of each I/O bank.

Step 5: Exporting the I/O Pinout

You can export the I/O port list and package pin information from PlanAhead software into a CSV format file, HDL header, or UCF file. The CSV file includes information about all of the package pins in the device, as well as design-specific I/O port assignments and their configurations. The package pin section of the list makes a great starting point for defining I/O port definitions in the spreadsheet.

You can also use the spreadsheet to automatically generate the PCB schematic symbols your group needs to begin the PCB layout. Sometimes, however, these symbols

are too large for the schematic; you'll have to break them up into several symbols. You can do this quite effectively using the interface groups created in PinAhead.

Providing this I/O pinout configuration in the form of a schematic symbol gives PCB designers a good foundation to start developing an effective PCB layout. Because you developed the initial pinout with the PCB interfaces in mind, it should be fairly close to the final pinout configuration.

And if you do need to swap pins during layout to facilitate routing, you should be able to make those changes easily – because more likely than not the pins that require swapping are conveniently located within the I/O banks. And the methodology shouldn't affect the FPGA design too drastically. You can transfer revised pinout spreadsheets or UCF files between your PCB and FPGA designers to keep in sync with any modifications you've made.

You may also wish to tie the unused pins or certain configuration pins to either VCC or GND (ground) to help with signal noise concerns and facilitate proper FPGA functionality. At Xilinx, we are currently working on a function for the next version of PinAhead that will provide an interface for FPGA designers to specify these pins and include the pins' connections in the output CSV spreadsheet. This will allow PCB designers to more easily identify pins and connect them appropriately.

Moving forward, as FPGAs employ more complex functions and use more advanced packages, devising a solid FPGA and PCB pinout methodology will become imperative. Although we think you'll find PinAhead Lite a great help in executing your co-design-savvy pinout strategies today, we are already working on improvements to the tool to better help you tackle pinout challenges and get the job done. 🌟