# Design elegance

FPGA-PCB codesign; a 21st Century approach to integrating fpgas into the pcb design process. By **Hemant Shah**.

Integrating advanced fpgas on a pcb is becoming increasingly challenging, with issues including generating optimal fpga pin assignments that do not add layers to a pcb or increase the time required to integrate the fpga with the pcb design. Because of this, fpga designers, schematic engineers and pcb designers struggle to create fpga pin assignments that meet the goals for the entire system.

With the increased capacity, capability and complexity in pin assignment rules, the time required to design in an fpga is increasing for most projects. With the traditional approach, designs are 'thrown over the wall', which only increases the number of iterations and the level of frustration. What makes it worse is that the processes are manual, increasing the chances of introducing errors that may not be discovered until the first prototype board comes back.

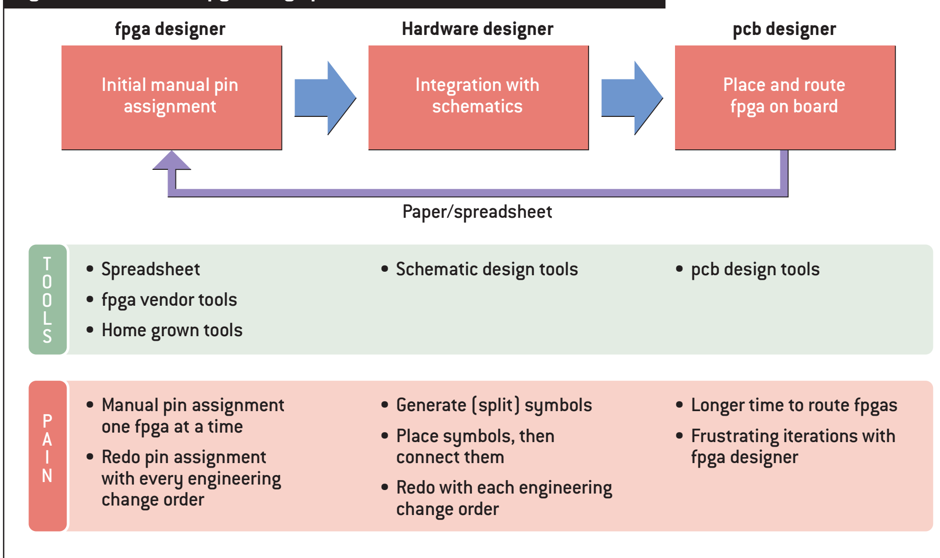



Fig 1: The traditional fpga design process is serial, manual and iterative

### Traditional manual approach

Many design teams use either the fpga vendor's tools or a home grown spreadsheet based solution for pin assignment and both approaches are manual (see fig 1). The fpga vendor's tools check pin assignments to ensure they comply with design rule checks (DRCs); the home grown spreadsheet based tool, meanwhile, needs to be updated for each new fpga that comes on the market. In both cases, the fpga designer or hardware designer works with one fpga at a time.

Shah: "Complex, high pin count fpgas require a 21st Century fpga-pcb codesign methodology."

The problem becomes more complex with multiple fpgas. If the relative placement (floorplan) of the fpgas on the pcb is not considered, the result will be poor pin assignment, which means longer routing and increased number of layers. If end product cost is an issue, there are likely to be multiple iterations as the pcb design team attempts to optimise the pin assignment for fpga timing and for pcb routing.

However, there is a third element that is usually not talked about: the process of integrating fpga symbols into the schematics that drive the pcb layout and routing. Symbols have to be created if you are using only a schematic capture tool – for example, not using table driven design creation tools. What is worse is that this integration – which creates multiple split symbols for a large pin count fpga, then connects the pins on the symbol in the schematics to other symbols – only serves one purpose; to provide connectivity for generating a netlist for board layout. What happens when there is an engineering change order (ECO) from the fpga designer or from the pcb designer? The hardware designer has to integrate the updated symbols and redo the connectivity all over again! This entire traditional process is iterative, time consuming, error prone and frustrating.

### There has to be a better way!

What is needed is a fpga-pcb codesign solution that leverages the information from fpga vendors about pin assignment rules (fpga I/O DRCs) from the relative placement, bga breakouts and connectivity of the fpga subsystem. Since many current fpgas come in large pin count packages, users also need a way to automate the pin assignments based on connectivity, fpga I/O DRCs and the relative placement of components that interface to the fpga(s).

This automatic pin assignment enables a couple of new things. Firstly, it shortens the optimum fpga pin assignment time, potentially from weeks to hours. Secondly, it allows the user to undertake architectural exploration and cost-performance trade off analyses, with the ability to substitute one fpga for another in hours. This architectural exploration is not possible with traditional methods.

It is clear that fpga-pcb codesign should integrate very well with all three domains – fpga vendor tools, hardware design (authoring) tools and pcb layout tools.

FPGA designers use fpga vendor tools to ensure timing closure with initial pin assignments or with any changes to the pin assignment, whether they are proposed by the fpga-pcb codesign tool or by pcb designers looking to tweak the initial pin assignment. The fpga-pcb codesign tool must understand the fpga's constraints, pin assignment rules and any additional signal integrity/power integrity rules that are specified by the fpga vendor. The tool must be able to optimise pin assignments based on constraints from all three domains – fpga design, hardware design and pcb routing.

The fpga-pcb codesign tool must accept constraints from and pass constraints back to the fpga vendor's tools. This bidirectional integration is necessary, especially for new interface IP – such as DDR2 and DDR3 – that fpga designers use from within the fpga vendor's tools. This integration was recently improved in Cadence Allegro FPGA System Planner.

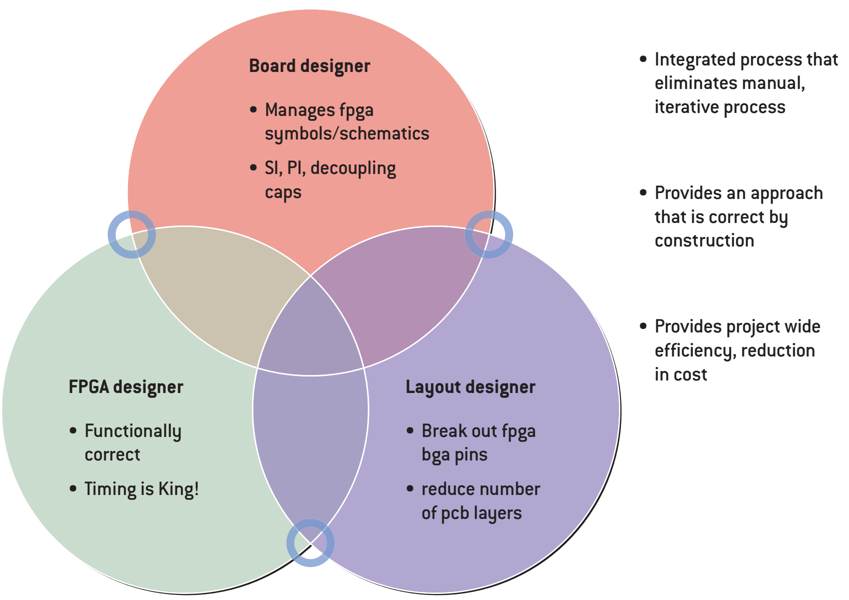
For integrating fpga connectivity with the rest of board – schematic integration – the fpga-pcb codesign approach generates the schematic symbols and their connectivity automatically. This makes engineering change order (ECOs) easy to manage. Consequently, a new schematic generation capability was added to Allegro FPGA System Planner in 2011

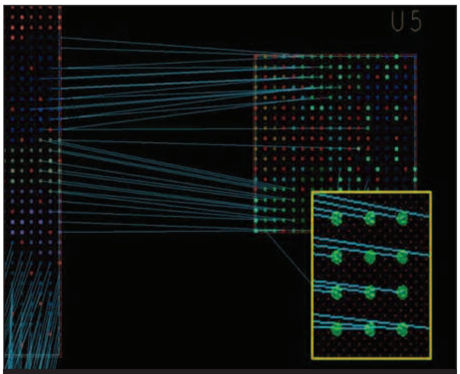
This codesign approach replaces a frustrating manual iterative methodology with a streamlined and efficient process (see fig 2). It provides a correct by construction methodology that reduces iterations, improves the routability of fpgas on the board and, potentially, reduces number of pcb layers. Best of all, ECOs are easy to implement with this approach; it is a simple matter to regenerate correct pin assignments and to make changes to the schematics.

ECOs are a fact of life in electronic design. The architecture could change during the implementation or the fpga designer could decide to swap fpgas for lower cost or to improve performance. There are many reasons for an ECO and an fpga-pcb codesign methodology that makes it easier to implement ECOs can greatly speed time to market.

Integration with the pcb layout tool should be such that the pcb designer can propose pin

**Fig 2: 21st Century fpga-pcb codesign – multiple designers, collaborative design**




*The tool suggests legal swap candidates*

swapping that is legal and within the bounds of the fpga designer's pin assignment intent. To accomplish this, the fpga-pcb codesign tool must tell pcb designers what the swap candidates are (see above). This eliminates unnecessary iterations between pcb layout and fpga design.

With this innovative approach, the number of iterations between pcb designers and fpga designers is reduced significantly.

**Summary**

FPGAs are becoming increasingly more complex to design and to integrate with the pcb design process. Meanwhile, traditional approaches to integrating today's complex fpgas are resulting in longer design in times and an increase in the number of iterations between pcb designers and fpga designers. Some designers are forced to choose between extending the project finish date or increasing the pcb layer count. FPGAs can be designed in more quickly, there can be fewer iterations and routing can be easier when using tools that understand the three domains that fpgas impact – the fpga's internal design, integration with schematics and pcb layout and routing.

Today's advanced, complex, high pin count fpgas require a 21st Century fpga-pcb codesign methodology that provides a correct by construction, automated flow for optimised pin assignment, automated symbol generation, easy schematic integration to shorten the design cycle, architectural exploration and a reduced number of iterations.

**Author profile:**

Hemant Shah is product marketing director with Cadence Design Systems (www.cadence.com).