Digital Design eSeminar Series Q&A: Dramatically Accelerate In-circuit Validation of Systems with FPGAs

The following Questions and Answers were created from the live eSeminar broadcast of May 18, 2004. You can view the archived eSeminar by going to **www.agilent.com/find/archives** and selecting it from the list of archived eSeminars.

Q: Is this dynamic probing, or is it just fixed probing with more signals?

A: I would describe this as the ability to switch between groups of signals. If you have sixteen pins of your FPGA that you want to dedicate, you're going to be looking at sixteen signals at a time or 32 signals if you use that pin compression mode. And then when you switch -- we say dynamic because you're doing this on the fly inside the FPGA to another set of sixteen signals and then another set of sixteen. It's dynamic because you don't stop the target; it's running, and you are incrementally measuring new groups of internal signals.

Q: Do you have to lay out connectors a certain way to use this solution?

A: No. All types of probing layouts will work if you already have completed board layout. You can still use the classic Mictor connector, and those are very popular; and you saw how the menu comes up with a picture of that. You can use a variety of different kinds of connections including Mictors, soft touch, termination adapters, even flying leads. So our goal was that you really can have a wide variety of choice.

Q: Let's say I decide what nets I want to see and I find a core that can switch around to these various signals. What do I have to do if I realize that I forgot some signals?

A: What you really have are options again. Let's say you define your 320 signals, 320 nets that you felt were important, like this last example where this designer had his 40 signals and his eight banks. And you say, "Oh, I completely forgot about some address bus that's really important to me as well." One of your options is you could use FPGA Editor from Xilinx and go ahead and route those out without doing a whole place and route of your design. Then if you go in and edit your CDC file that will allow those names to come on through to the logic analyzer. So that's certainly one of the approaches. The other is go ahead and just go back into core inserter of the ChipScope Pro and redefine your core. I think you saw it's a pretty simple process where you see the design; you click on the signals you want and assign those to banks. So you would do that -- now that would require that you go ahead and do a place and route again. That happens post-synthesis if you use the core inserter and it would take the time to place and route, but it's really easy enough that that would be a fine approach as well.

Q: How are the JTAG lines connected to the logic analyzer? Do we need extra hardware? Use the Xilinx parallel cable, or a special Agilent connector cable?

A: You're going to have a Xilinx cable that will connect to either the USB or the parallel output of your PC where you're running, in this case, the logic analyzer running the FPGA dynamic probe. All Agilent Windows-based logic analyzers have parallel and USB ports on the instrument. Plug the Xilinx cable into this port and then connect it to the JTAG pins on your PC board. There are several different types of cables from Xilinx that you can use; and those are about $100 each. So that's really how you make the connection to control the core by JTAG. And then, obviously, the signals that you're bringing out to measure, that's where you're going to use a Samtec, or a Mictor, or Flying Leads, or Soft Touch connectorless.

Q: How can I mix the Xilinx ILA cores with the ATC II cores?

A: You can certainly place both types of cores into your design. If you recall, the ILA core is the Xilinx core that works with the ChipScope Pro analyzer, versus the ATC II core, which would be used with the FPGA Dynamic Probe and a logic analyzer. If you're going to use the ILA cores, then you will use the Xilinx ChipScope Pro interface to look at those signals. If you are going to use the ATC II core, you'd use the logic analyzer with the FPGA Dynamic Probe. A way to use them together is to host the ChipScope Pro session on your logic analyzer. Both the ChipScope Pro application as well as the FPGA dynamic probe application can both talk to the same JTAG cable on the logic analyzer and interface with the same scan chain.

Q: Can you look at a combination of signals from various banks, or do you get all sixteen signals from one bank or none?

A: You're going to be seeing one bank at a time. So as we talked with the example of the sixteen pins, you will see sixteen signals mapped in. Now you might see many other signals that are part of the rest of the system, but as far as those signals coming from the FPGA, you'll see that one bank worth. And that's why you have to make some trade-off decisions. If you looked at that table closely, you're trading off how many resources are required if you want to use that pin compression mode -- and that might be a great choice. Okay, you've got sixteen pins, but now I can see 32 signals at once and be able to have the visibility that you want.

Q: Do we need Xilinx ChipScope Pro tool also if we use your tool?

A: To use the FPGA Dynamic Probe you need Xilinx Core Inserter, which is packaged in ChipScope Pro. For the whole solution, here's what you need. You're going to need a Windows XP-based Agilent logic analyzer; either the 16900 System analyzers or 1680 and 1690 bench tops support this application. You need to have the ChipScope Pro software from Xilinx. So that you get from Xilinx; you can get it from the web, there's various ways that you get that. You're going to need a Xilinx JTAG cable like we talked about, to make the JTAG connection. And then you're going to need the FPGA Dynamic Probe application that you get from Agilent. So really, there are several pieces from Agilent and Xilinx that all work together.

Q: Would I normally remove the debug core prior to shipping the final product?

A: As I have talked to many designers, traditionally, they're going to leave them in. They're not that large, it's not like they're using up resources that would allow them to have their design fit or not; so it's just really easiest to leave them in. Plus, if you come back later with some problem in the field, you've got a very simple way to hook right up to a shipping board and try to see what's going on inside. So I wouldn't say there's a lot of benefits to ripping all that back out. You've got it in there, everything's working, you've debugged with it -- leave it in. Some people would rather take it out. One nice thing about inserting the core after synthesis with the core inserter, you haven't really introduced debug into your high-level code, your Verilog or HDL. That stays pretty clean; this is put in separately as you merge net lists. So it's a nice way to have it there in the future if you need it. If not actively being used for debug, the core stays powered off.

Q: How much does it cost for this solution, both hardware and software?

A: The ChipScope Pro is $695 and it's a yearly license from Xilinx. The cable is about $95. The FPGA Dynamic Probe application is $995 per year with a license. And then your logic analyzer can be everything from a hosted type of bench top analyzer, which starts around $7,000, on up to a larger system analyzer with many, many channels; those can be, you know, quite a range -- $20,000, $80,000. So those are the kinds of things that you put all those together and then you'll be able to have this solution.

Q: Is there a mechanism to bring the state analysis clocks through the FPGA signal pins being monitored, or do they need to be available external to the FPGA?

A: If you're going to use the state cores, you will define which signal, which net is going to be your system clock, if you will. So let's say in that first example, we had four banks of sixteen signals each. You're going to define a main clock for that core. Now if you've got multiple clock domains, what you're probably going to want to do is put in several cores -- each that would be fed by the state clock from your FPGA for that clock domain -- or use a timing core. And now you can be looking at multiple clock domains -- because the analyzer's basically sampling it at a very fast rate and then you can be able to see what's happening between the domains. That's the way that works. In ChipScope

Pro core inserter you tell it, "I want this to be the driving clock for this core."

Q: Can you do this with other manufacturer's FPGAs?

A: We've worked closely with Xilinx in the creation of this capability. So this works with Xilinx Spartan III, Virtex-II, Virtex-II Pro FPGAs and that's what this is able to work with; it does not work with other vendors at this time.

Q: Is there a demo lab available for this solution? I would benefit from learning by doing.

A: In probably July 2004, there will be about sixteen labs -- they're called Xilinx RocketLabs -- around the world. I think six of them in the United States, the rest international (Europe, Japan, Asia). The labs will have a system that we're talking about that has this FPGA Dynamic Probe capability. So the idea of that is that you can visit -- these will be at Xilinx facilities. And you'll be able to either work through this to see how it works on kind of a demo board thing, or do some more lab-type things with the kind of designs you want to do. So, yes; the answer is, these will be available -- and also our Agilent applications engineers are equipped to demonstrate this application.

Q: Does the pin compression place a performance limitation on the FPGA?

A: Yes, TDM affects the timing budget. This is because the TDM feature will use the DDR flip flops to register the core outputs on the rising and falling edge of the state core clock. As a result of this the period constraint is cut in half for the IO flip flops (DDR flip flops). However this additional constraint is not too severe on your design. This is because the ATC2 is a piped circuit that isolates your probed signals from these IO flip flops through intermediate registers.

One other factor you must consider with TDM is the output signal quality. Since TDM doubles the data bandwidth, the core output buffers and the printed circuit (PCB) traces must be tested to ensure signal integrity on the logic analyzer trace connector. With TDM you might have to experiment with different output buffers to find one that works best on your PCB. So if you're running your FPGA at 100 MHz, the I/O will run at 100 MHz with data on both edges, or 200 Mb/s. The logic analyzer has a special mode to decompress the data The way the logic analyzer works is, on the rising edge of clock, it acquires one signal and on the following edge of clock, it acquires the other. So then you've got a data rate that's running -- you know, it's a double data rate of your frequency of your clock and your FPGA. The overall performance will be driven by the FPGA tools, which are trying to meet timing constraints. What I would say with these solutions, your best approach is to insert the core. Go ahead and do your place and route and see how that ends up turning out in terms of the timing constraints. And from our experience, especially with the timing core and the basic state core, it should run right at the full speed of your FPGA. With the time-division multiplexing, there you might have to experiment a little bit to see how the constraints work out.

Q: Does it work for asynchronous signals too?

A: That's where you'd use the timing core. The timing core lets you see glitches that have duration less then the clock period. You can sample, let's say, 4 GHz-timing rate, very fast, and then have all these signals coming out at you. Even if they're in multiple or asynchronous types of situations, it is just looking for those signals and using the logic analyzer to determine when to sample. So that's what you'd use is the timing core for that.

Q: Are there any special design considerations I must make if I know I want to use these kinds of measurement cores?

A: There are several considerations you will find useful. One consideration is thinking about the widest bus they may want to measure internal see her FPGA. This will determine the width of your debug port. Another consideration is the impact on your design performance in size. The best way to manage this is to run a place and route with the core the design and then you can determine the exact impact.

Q: How will inserting a core effect the performance of my design? Will this lower my maximum clock speed?

A: In general, it shouldn't. The best way to check is to insert a core into your design and run place and route. You will then know if the addition of the core had any impact on your overall design performance. The timing core is a special case that will not impact performance. Since the timing core does not register inputs or outputs, it does not impose any additional timing constraints and inputs are designated as false paths. It will run at the full speed for sure.

Q: Can I get a copy of the presentation?

A: Yes, go to eSeminar archive. When the window opens there will be a row of button at the bottom of the window, such as “Enlarge Slides”, “Download Slides”, “Resources”, and “Feedback Form”. If you click on Download Slides, you can get a copy of the presentation in PDF format. If you click on Resources, the Agilent Resource page will be displayed, where you can find many valuable links for more information on the subject. There is also a Digital discussion forum (**www.agilent.com/find/forums**) if you would like to pose/review other questions.

Agilent Technologies