

In-circuit Debug of FPGAs

Application Note 1457

By Joel Woodward, Agilent Technologies

FPGAs are everywhere, but debugging an FPGA-based system can be frustrating due to lack of adequate internal visibility. With bigger FPGAs that often contain whole systems, visibility becomes a huge issue. What are the tools that can be used for taking internal FPGA trace measurements and what techniques exist to maximize internal visibility using a fixed number of pins?

FPGA designers have two methods of taking internal trace measurements:

1. Routing nodes to pins and using a traditional external logic analyzer or scope.
2. Inserting a logic analyzer core into the FPGA design and routing out via JTAG, trace capture stored using internal FPGA memory.

Logic Analysis

Early in the design, FPGA developers make critical decisions, consciously or unconsciously, that determine how debug-able their design will be.

The most common method of getting internal FPGA visibility is with a logic analyzer. Internal nodes of interest are routed to pins that the analyzer probes. This approach offers deep memory traces where the cause of a problem and the effect can be separated by large time intervals and the resulting measurements can be time-correlated with other system events.

Traditional logic analysis provides both state and timing modes, so data can be captured asynchronously or synchronously. In timing mode, a designer can look at the relationship of signal transitions relative to one another. Using timing mode, logic analyzers gracefully measure asynchronous events that might escape simulation such as the FPGA interaction with other parts of the system.

In state mode, a designer has the ability to view buses relative to a state clock. State mode is particularly useful in debugging data path design when the value of the bus over time is important.

Effective real-world measurements require up-front planning. Early in the FPGA development process, designers specify a specific number of pins that will be dedicated to debug. This specification occurs early—and without the knowledge of what specific debug issues will present themselves during development. Traditional logic analyzers can only view signals routed to pins. The number of pins available for debug may not provide adequate visibility for the problem at hand, delaying project completion.



Logic Analysis: Minimizing Number of Pins Dedicated to Debug

One method of retaining internal visibility while reducing the number of pins dedicated for debug is inserting a switch mux into the design. (See Figure 1.) For example, while 128 internal nodes may need to be viewed when the FPGA design goes in circuit, it may be sufficient to trace 32 channels at a time. In this case, a mux can be implemented in the FPGA design to route out 32 nodes at a given time. To program the mux, a designer may switch which signals are routed by downloading a new configuration file, using JTAG, or via control lines on the mux. Test mux insertion must be carefully planned during the design phase. Otherwise a designer can end up in a situation where the nodes needed for debugging may not be accessible simultaneously.

A second method of minimizing the number of pins dedicated to debug is time-division multiplexing (TDM). TDM muxing, more common in prototyping where multiple FPGAs are used to prototype a single ASIC, can also be used to minimize pins dedicated to debug. This technique applies best when dealing with slower internal circuits.

Assume a 50 MHz design (20 ns between clock edges) with an 8 bit bus that needs in-circuit visibility. Using a 100 Mhz clock one can sample the lower 4 bits during the first 10 ns and the upper 4 bits during the second 10 ns. During each 20 ns cycle, all of the 8-bit debug information can be captured using only 4 pins. After a trace has been captured, combining consecutive 4-bit captures provides reconstruction of the 8-bit trace.

There are a couple of downsides of TDM muxing. If capturing the trace with a traditional logic analyzer, triggering becomes more complex and error prone. For example, triggering on an 8-bit pattern involves setting the logic analyzer trigger to look for a particular 4-bit pattern followed sequentially by another specific 4-bit pattern. However, the logic analyzer doesn't know which 4-bits is the start of the 8-bit group and thus may trigger on a condition that matches the trigger setup—but not what the user had intended to trigger on.

As with all synchronous acquisition methods, the resulting measurement is cycle accurate. The designer loses timing relationship information between clock cycles. The TDM compression ratio is typically limited by single-ended pin speed and the acquisition speed (state mode) of the logic analyzer. For example, if the maximum single-ended pin speed is 200 Mhz and internal circuits are running up to 100 MHz, a 2:1 pin compression ratio is the maximum that can be achieved.

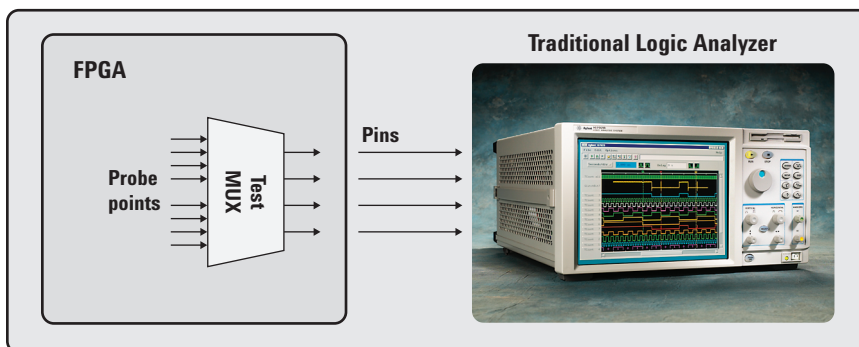


Figure 1. Insertion of a test mux gives the designer the ability to route out a subset of internal signals while minimizing pins dedicated to debug. The resulting trace is captured with a traditional logic analyzer such as the Agilent 16702B.

Mixed-Signal Oscilloscope

Designers can also connect debug pins to scopes.

An MSO (Mixed-Signal Oscilloscope) can serve as a basic, easy to use tool for observing FPGA activity as well as observing signals on the FPGA boundary. Because an MSO offers 4 analog channels plus another 16 digital channels with deep memory capture, the MSO provides more advanced triggering and capture capabilities. The analog channels of an MSO or a standard oscilloscope offer insight into FPGA signal parametrics.

MSO limitations for FPGA debug include a maximum of 20 channels visibility and the inability of triggering or capturing trace synchronously. Depending on how difficult the debug situation, the MSO can serve as a starting point, and a logic analyzer can be brought in when more complex triggering, greater channels, or other logic analyzer capabilities are needed.

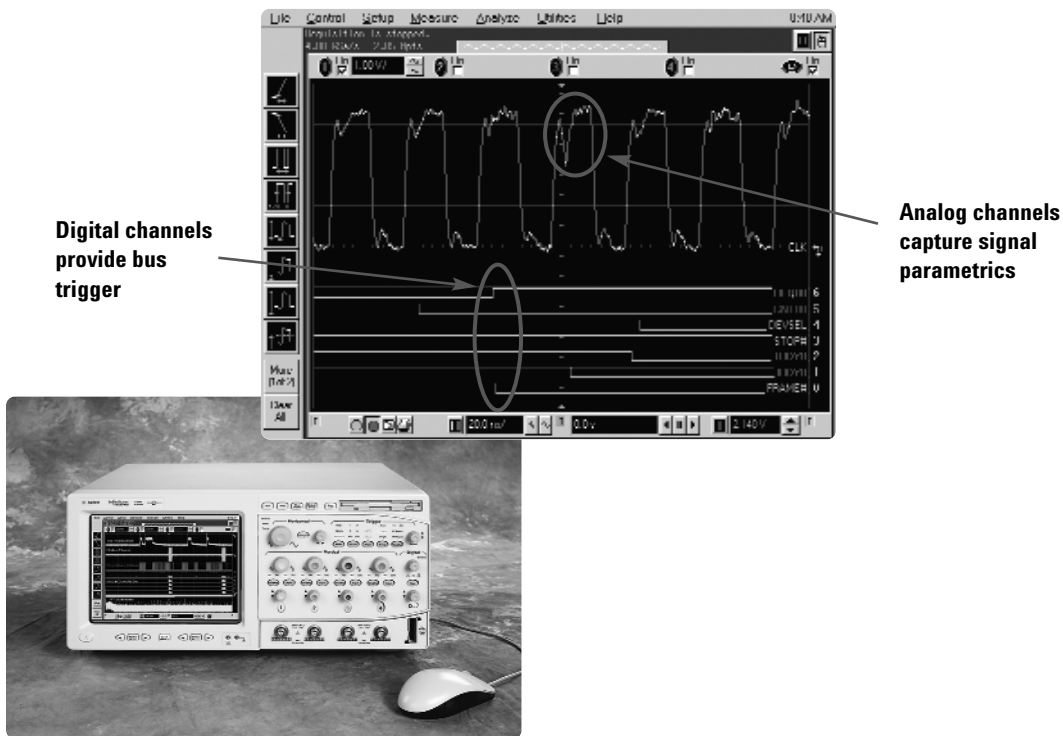


Figure 2. Mixed signal oscilloscopes provide powerful FPGA debug capabilities by combining 16 digital channels with up to 4 analog channels.

Logic Analysis Cores

As a given FPGA design matures, it may receive enhancements and changes. Over time the pins dedicated initially for debug are consumed by design enhancements. Or the design may have initially been pin limited. Another debug technology provides value in these situations.

Most FPGA vendors now offer internal logic analysis cores. (See Figure 3.) These pieces of IP can be inserted into the FPGA either pre- or post-synthesis. The core contains a trigger circuit and resources used to set up a measurement as well as internal RAM for trace storage. Inserting the logic analysis core into the design changes the timing of the design and hence most designers will leave the core in the design permanently.

The core is accessed via JTAG for in-circuit configuration as well as to pass captured data to a PC for viewing.

When using an internal logic analysis core, it is important to choose an FPGA with sufficient resources to allow the core and the design to route appropriately. Vendors typically specify the resources used by specific logic analysis cores. In the Xilinx case, one can divide the number of slices the core takes by the number of slices available in a particular device. Typically, if this ratio is less than 5%, the FPGA will have adequate resources to include the core.

Logic analysis cores have three main advantages.

1. They use no incremental pins. They are accessed via the dedicated JTAG pins already on the FPGA. This method of debug can be used for internal visibility even when no additional pins are available.
2. Simple probing. Probing involves routing nodes to the input of the internal logic analyzer. There is no need to worry about how to connect to the board and no signal integrity issues in getting valid information.

3. Logic analysis cores are inexpensive, logic analysis IP generally available for less than \$1,000.

There are three primary trade-offs of using internal logic analysis cores.

1. The size of the core limits use of the core to large FPGAs. Additionally, because internal FPGA memory is used for trace, trace depths tend to be shallow.
2. Designers must give up internal memory for debug, memory they likely want to be able to use for the design.
3. Internal logic analyzers operate exclusively in state mode. They capture data synchronous to a specified clock and do not provide signal-timing relationships.

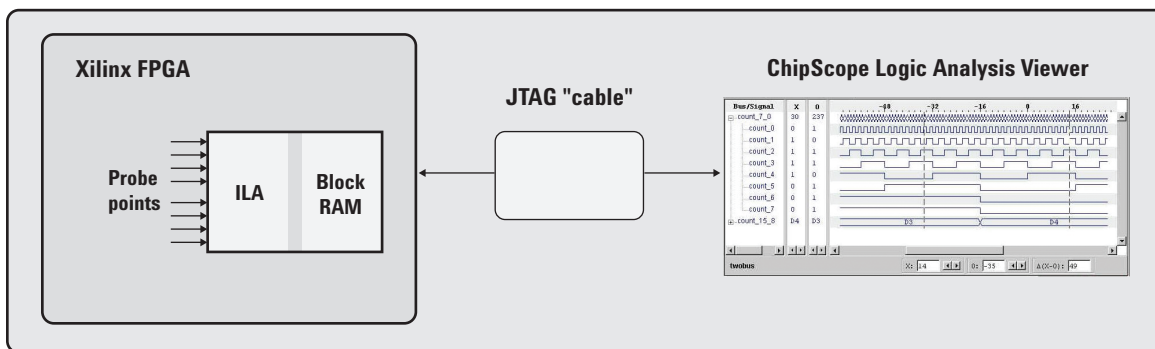


Figure 3. On-chip logic analyzer requires no incremental pins. The logic analysis configuration is downloaded over JTAG. The host computer uploads the resulting trace from internal FPGA memory to a hosted logic analyzer viewer. An example of this technology is Xilinx ChipScope Pro.

Hybrids

Some FPGA vendors have begun working with traditional logic analysis vendors to combine technologies. (See figure 4.) As an example, Agilent and Xilinx co-developed 2M state deep memory for Xilinx's ChipScope Pro.

The solution utilizes an internal logic analysis core for triggering. Once the core's trigger condition is met, the core passes trace

information to pins. The pins connect via a mictor connector to a small external trace box. The solution can incorporate TDM muxing to reduce the number of pins dedicated to debug. Muxing compression can be 2:1 or 4:1, depended on the speeds of the internal circuit. Because trace is not stored internally, the IP core is smaller than a stand-alone ILA core with trace memory.

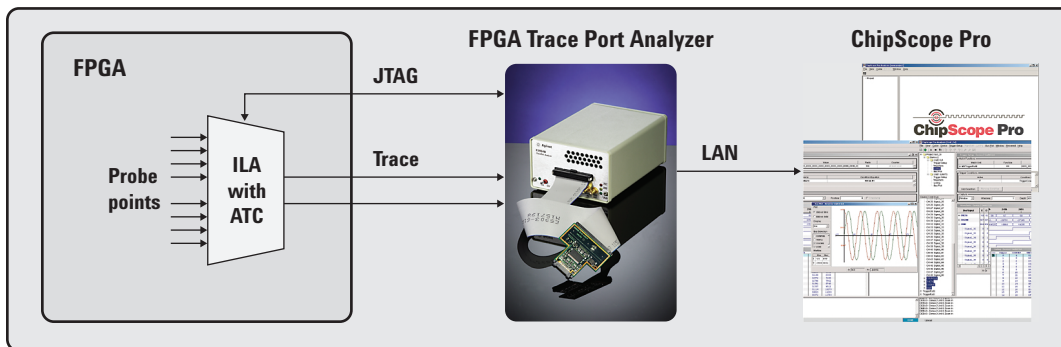


Figure 4. The first commercial instance of internal logic analysis with external storage comes from Agilent and Xilinx. They teamed to develop deep memory for ChipScope Pro. This technology offers a lighter logic analysis core and pin minimization using TDM multiplexing.

How to Decide

Traditional logic analysis and core-based logic analysis technologies are both useful. Upfront consideration of several factors will help determine when each is a better fit for your debug needs. Here a few questions to help guide the decision of which will be most effective.

1. What type of debug problems do you expect to encounter? Simpler problems can be readily found with internal logic analyzers while traditional logic analyzers excel with sophisticated bugs.
2. In addition to state mode, will you need to capture timing information? If so, a traditional logic analyzer uniquely provides this value.
3. How deep of traces will be needed? Traditional logic analyzers go capture trace up to 64 M on each channel while internal logic analysis cores are better suited for shallow trace.
4. How many pins can be dedicated for debug? The smaller the number, the better the fit with an internal logic analyzer.
5. How much money does the team have to invest for new tools? While traditional logic analyzers start at \$6 K for 32 channels, internal logic analyzers and accompanying waveform viewers start at less than \$1 K.
6. What impact on the FPGA design is the team willing to tolerate? Internal cores work on large FPGAs exclusively and change the timing of the design. Routing signals out to traditional logic analyzers has less of an effect on the design and works with all sizes and types of FPGAs.

As FPGAs continue to incorporate more and more capabilities, the need for effective debug tools will become more critical. Upfront planning for internal visibility allows teams to incorporate the right debugging strategy to get their designs working quickly.

Related Literature

Publication Title	Publication Type	Publication Number
<i>Agilent Technologies 16700B and 16702B Logic Analysis Systems</i>	Brochure	5988-8435EN
<i>Using the 16700 with Xilinx ChipScope ILA</i>	Application Note	5988-0643EN
<i>Agilent 1680 and 1690 Series Logic Analyzers</i>	Brochure	5988-2675EN
<i>Probing Solutions for Logic Analyzers</i>	Product Overview	5968-4632E
<i>Deep Storage with Xilinx ChipScope Pro and Agilent Technologies FPGA Trace Port Analyzer</i>	Product Overview	5988-7352EN
<i>Infiniium 54830 Series Oscilloscopes</i>	Product Overview	5988-3788EN
<i>Digital and Mixed Signal Oscilloscopes</i>	Selection Guide	5988-8460EN

Agilent Technologies' Test and Measurement Support, Services, and Assistance

Agilent Technologies aims to maximize the value you receive, while minimizing your risk and problems. We strive to ensure that you get the test and measurement capabilities you paid for and obtain the support you need. Our extensive support resources and services can help you choose the right Agilent products for your applications and apply them successfully. Every instrument and system we sell has a global warranty. Support is available for at least five years beyond the production life of the product. Two concepts underlie Agilent's overall support policy: "Our Promise" and "Your Advantage."

Our Promise

Our Promise means your Agilent test and measurement equipment will meet its advertised performance and functionality. When you are choosing new equipment, we will help you with product information, including realistic performance specifications and practical recommendations from experienced test engineers. When you use Agilent equipment, we can verify that it works properly, help with product operation, and provide basic measurement assistance for the use of specified capabilities, at no extra cost upon request. Many self-help tools are available.

Your Advantage

Your Advantage means that Agilent offers a wide range of additional expert test and measurement services, which you can purchase according to your unique technical and business needs. Solve problems efficiently and gain a competitive edge by contracting with us for calibration, extra-cost upgrades, out-of-warranty repairs, and on-site education and training, as well as design, system integration, project management, and other professional engineering services. Experienced Agilent engineers and technicians worldwide can help you maximize your productivity, optimize the return on investment of your Agilent instruments and systems, and obtain dependable measurement accuracy for the life of those products.



Agilent Email Updates

www.agilent.com/find/emailupdates

Get the latest information on the products and applications you select.

Agilent T&M Software and Connectivity

Agilent's Test and Measurement software and connectivity products, solutions and developer network allows you to take time out of connecting your instruments to your computer with tools based on PC standards, so you can focus on your tasks, not on your connections. Visit www.agilent.com/find/connectivity for more information.

By internet, phone, or fax, get assistance with all your test & measurement needs

Online assistance:

www.agilent.com/find/assist

Phone or Fax

United States:

(tel) 800 452 4844

Canada:

(tel) 877 894 4414

(fax) 905 282 6495

China:

(tel) 800 810 0189

(fax) 800 820 2816

Europe:

(tel) (31 20) 547 2323

(fax) (31 20) 547 2390

Japan:

(tel) (81) 426 56 7832

(fax) (81) 426 56 7840

Korea:

(tel) (82 2) 2004 5004

(fax) (82 2) 2004 5115

Latin America:

(tel) (305) 269 7500

(fax) (305) 269 7599

Taiwan:

(tel) 0800 047 866

(fax) 0800 286 331

Other Asia Pacific Countries:

(tel) (65) 6375 8100

(fax) (65) 6836 0252

Email: tm_asia@agilent.com

Product specifications and descriptions in this document subject to change without notice.

© Agilent Technologies, Inc. 2003

Printed in USA May 1, 2003

5988-9433EN

