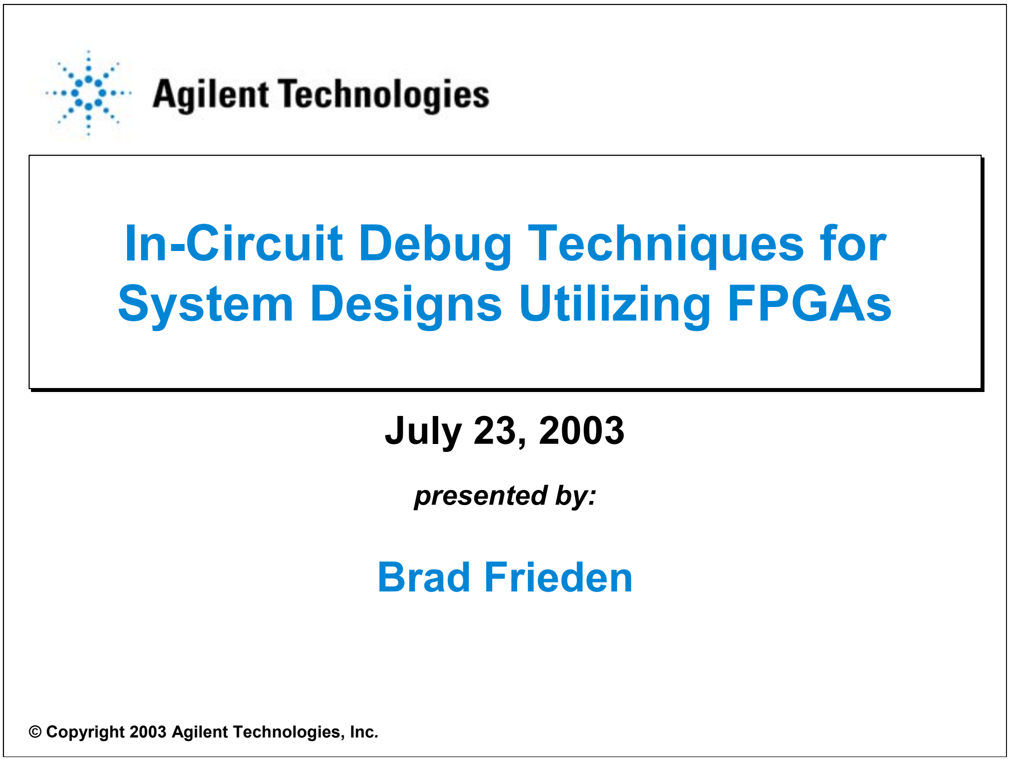


Agilent Technologies

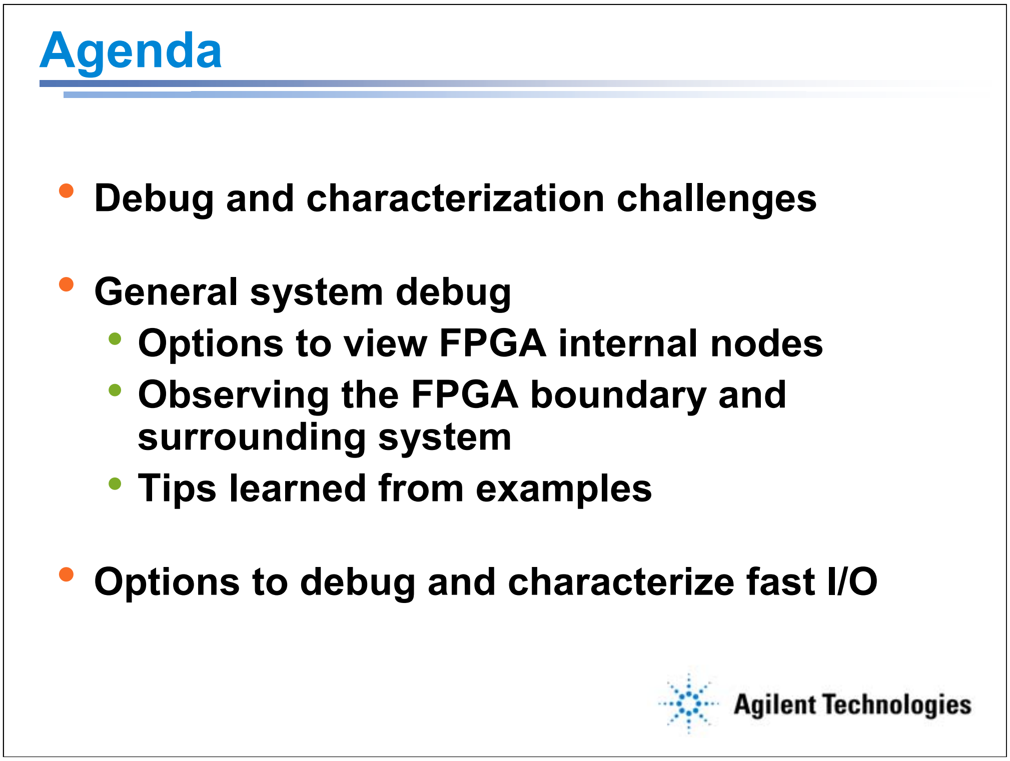
In-Circuit Debug Techniques for System Designs Utilizing FPGAs

July 23, 2003

presented by:

Brad Frieden

Welcome.



Agenda

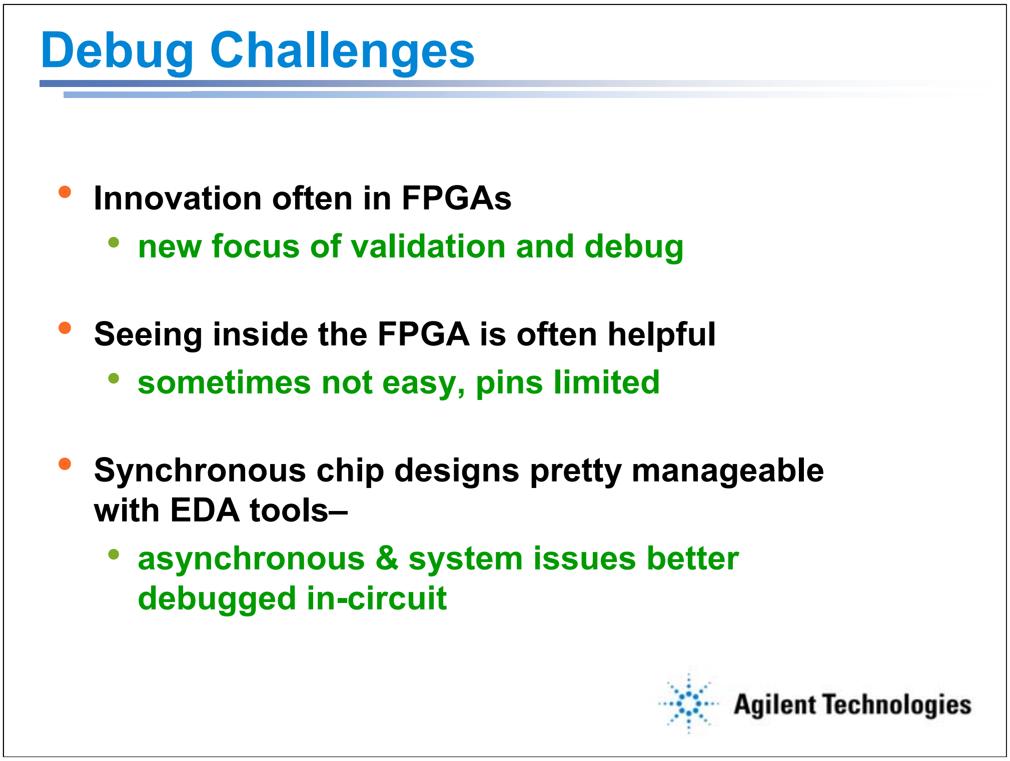
- Debug and characterization challenges
- General system debug
  - Options to view FPGA internal nodes
  - Observing the FPGA boundary and surrounding system
  - Tips learned from examples
- Options to debug and characterize fast I/O

Agilent Technologies

Agenda

Welcome to Agilent's eSeminar on debug methodologies for FPGA-based systems. Today, we'll look at the nature of such systems and how that presents debug challenges. Next, we'll consider the ways that one can get access to internal FPGA signals as well as other system signals. Then we'll spend most of our time looking at real-life design examples and how designers debugged problems in those systems. Based on what we learn from these cases, a number of helpful approaches and tips become evident and we will discuss those. Finally, we'll also consider options related to high-speed I/O.

Let's begin by considering some of the unique challenges related to FPGA system debug.



Debug Challenges

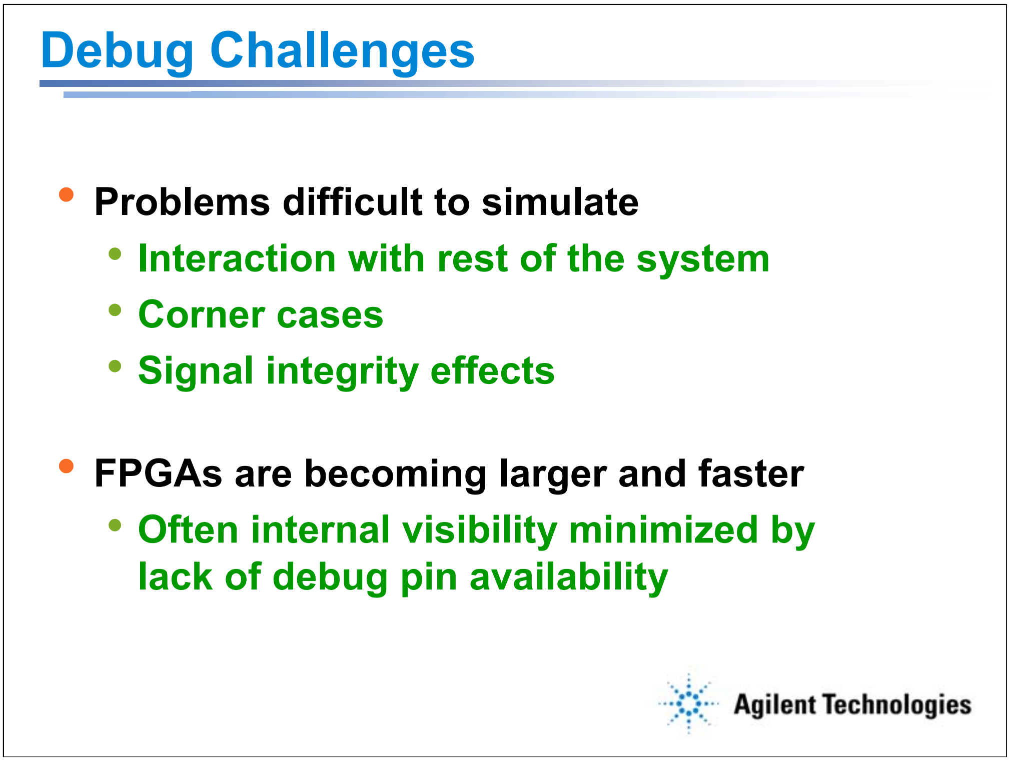
- Innovation often in FPGAs
  - new focus of validation and debug
- Seeing inside the FPGA is often helpful
  - sometimes not easy, pins limited
- Synchronous chip designs pretty manageable with EDA tools–
  - asynchronous & system issues better debugged in-circuit

Agilent Technologies

When you look to the nature of the F P G A systems, there are a number of factors that can make in circuit debug a challenge. First of all a good portion of the innovation is what is placed inside the FPGA, so the FPGA ends up becoming a focus of the debug and validation effort.

It is often very important to be able to see inside the FPGA but often that is easier said than done because the external pins are often not available for debug or only a limited number are available.

EPA tools work very well for synchronous designs but asynchronous situations, like crossing time domains can become quite tricky.



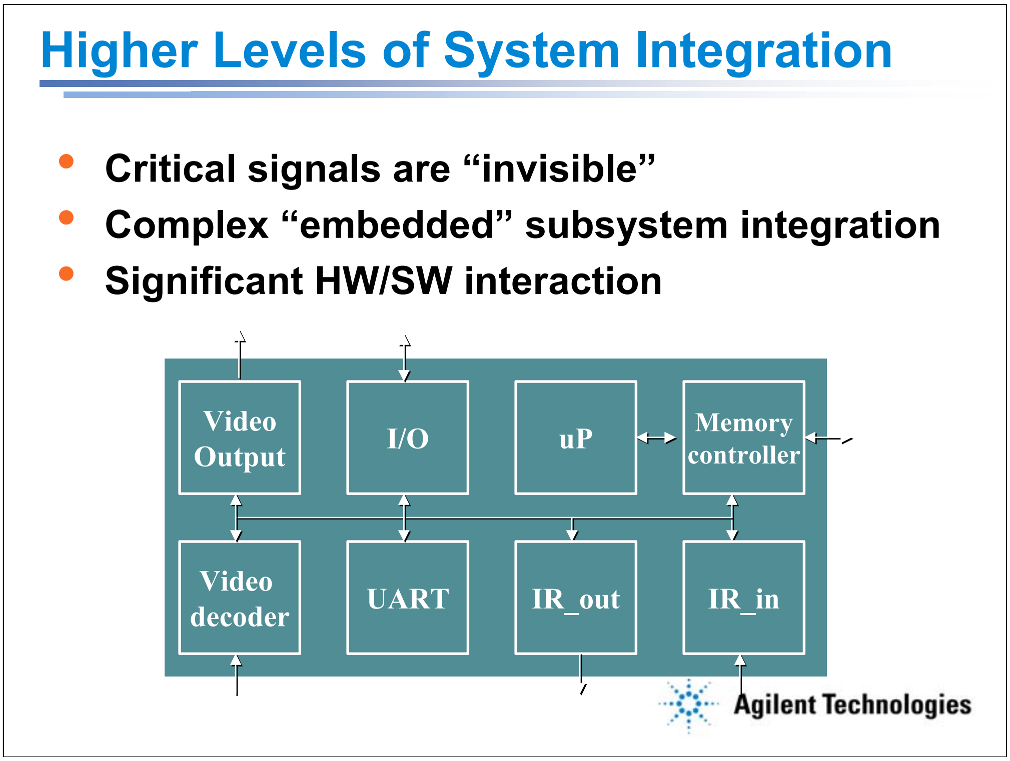
Debug Challenges

- Problems difficult to simulate
  - Interaction with rest of the system
  - Corner cases
  - Signal integrity effects
- FPGAs are becoming larger and faster
  - Often internal visibility minimized by lack of debug pin availability

Agilent Technologies

With some of the latest FPGAs is it is now possible to include large IP blocks such as microprocessors. Some of the IP, the hard cores, where Xilinx provides a black box functionality, does not allow you to look inside, thus limiting your visibility. With soft cores that are synthesized from R T L code, it is possible to look inside.

The size of available FPGAs is getting larger and the possible speeds are increasing. Even though the devices are larger, the designs are now larger too and can still find yourself pin limited.



Higher Levels of System Integration

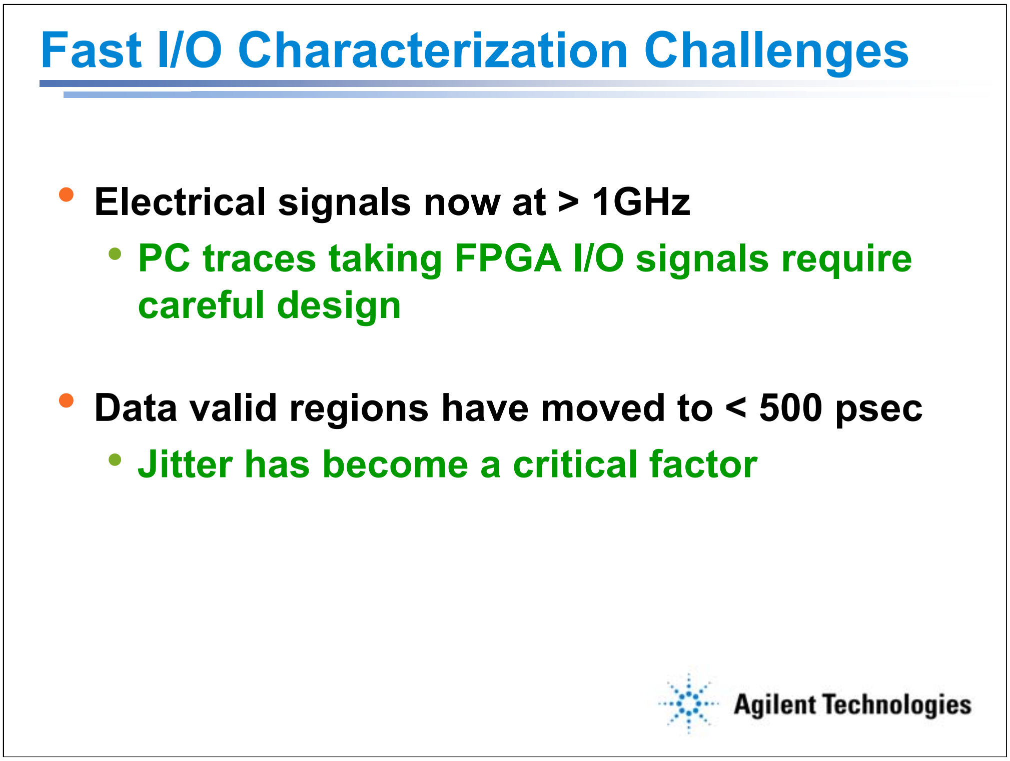
- Critical signals are “invisible”
- Complex “embedded” subsystem integration
- Significant HW/SW interaction

This increasing complexity presents some new challenges for debug and system integration.

Critical signals are buried deep and not naturally visible.

Entire subsystems are internal to the FPGA, limiting or complicating system integration & debug.

System complexity increases dramatically; integration challenges are higher than ever.

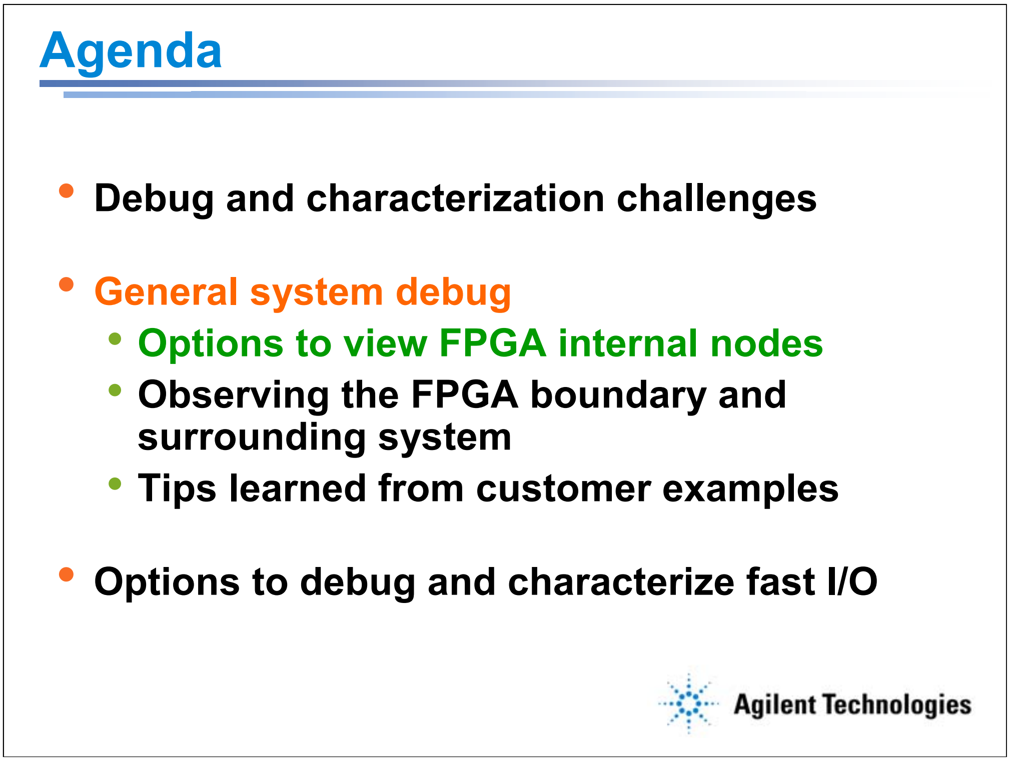


Fast I/O Characterization Challenges

- Electrical signals now at > 1GHz
  - PC traces taking FPGA I/O signals require careful design
- Data valid regions have moved to < 500 psec
  - Jitter has become a critical factor

Agilent Technologies

When it comes to Fast I/Os, especially those running at GHz rates, PC board traces are essentially acting as transmission lines. Part of the system design involving FPGAs with fast I/Os is the careful design of the signal path.

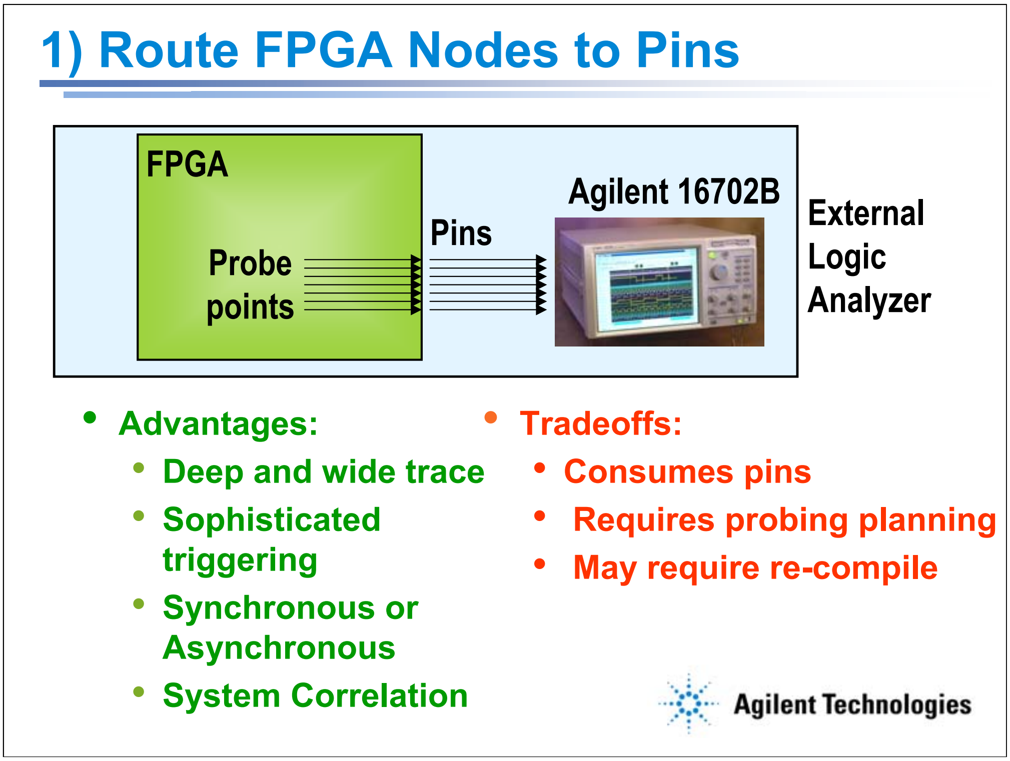


Agenda

- Debug and characterization challenges
- General system debug
  - Options to view FPGA internal nodes
  - Observing the FPGA boundary and surrounding system
  - Tips learned from customer examples
- Options to debug and characterize fast I/O

Agilent Technologies

Lets now turn to general system debug and the main ways it is possible to see internal nodes inside an F PGA



1) Route FPGA Nodes to Pins

FPGA
Probe points
Pins
Agilent 16702B
External Logic Analyzer

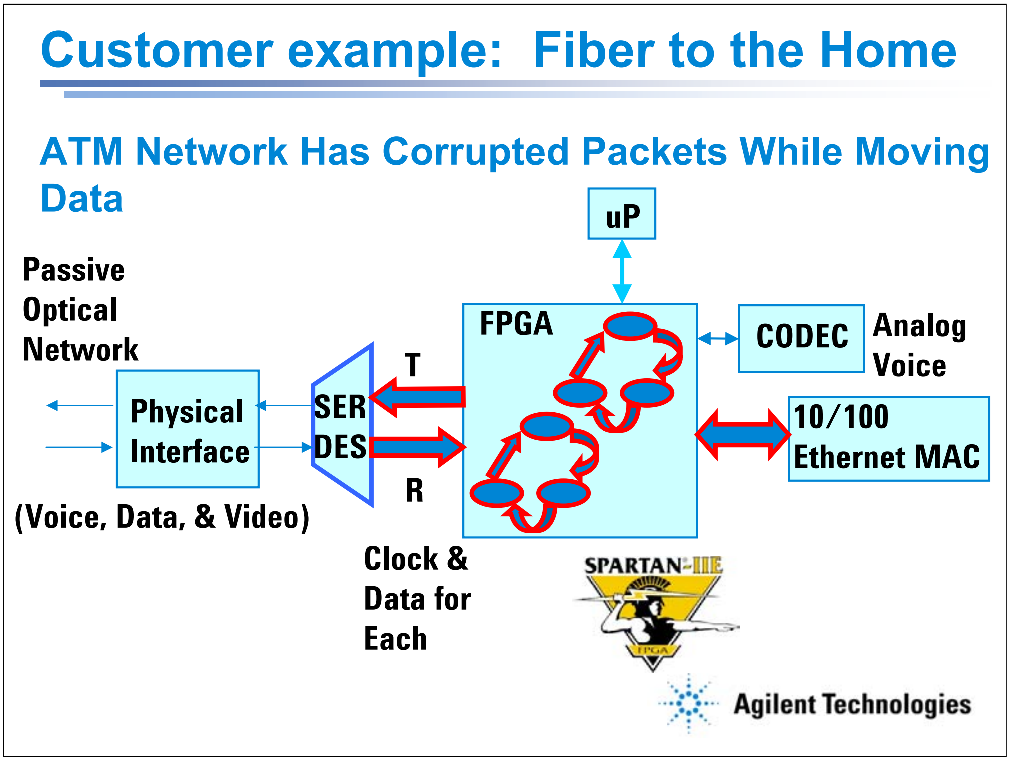
- Advantages:
  - Deep and wide trace
  - Sophisticated triggering
  - Synchronous or Asynchronous
  - System Correlation
- Tradeoffs:
  - Consumes pins
  - Requires probing planning
  - May require re-compile

Agilent Technologies

The first approach is pretty basic—just route out the nodes of interest directly to external pins on the FPGA and look at them with a logic analyzer. With this method come a number of advantages surrounding the debug features of an external logic analyzer such as deep memory and sequence triggering. It's also very straightforward to correlate internal FPGA signals with other signals captured from the target system.

Of course, the big issue can be how many pins are available for this. Certainly, this approach also carries with it a moderate investment. However, the logic analyzer is used for many other general purpose applications.

It's best to determine early in the development process how many pins you want to dedicate to debug.



Customer example: Fiber to the Home

ATM Network Has Corrupted Packets While Moving Data

Passive Optical Network

Physical Interface

(Voice, Data, & Video)

SER DES

T

R

Clock & Data for Each

uP

FPGA

CODEC

Analog Voice

10/100 Ethernet MAC

SPARTAN-IIE FPGA

Agilent Technologies

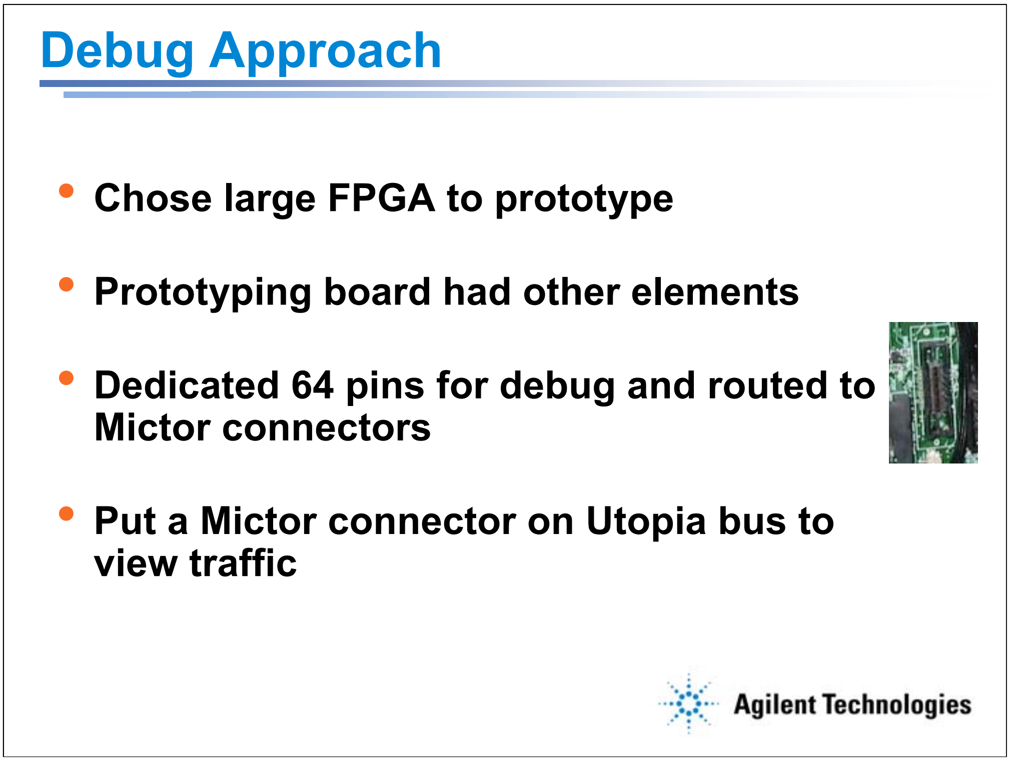
Let's look at an engineering example where this approach was used. Here a designer was experiences problem with a Fiber to the Home ATM Network that would reside at the customer location. This was the part of a system which connected to a passive optical network and then pulled out voice, data, and video into standard formats for use at the customer site. The system was experiencing corrupted packets while moving data. Some of the system characteristics were:

ATM network bringing fiber into a home and offering voice, video and data capability

The FPGA is in the data path and takes in raw data from fiber optic clock and data recovery circuits. The FPGA served as a packet processing engine. Multiple state machines, triggered by cells coming through, would look at cell headers and determine what kind of service was being processed. They would then perform decoding, buffering and formatting operations. The state machines had to be "perfectly timed".

Puts out Ethernet frames to the 10/100 Ethernet parts

Are a number of Utopia buses all over and "sort of Utopia" buses for proprietary chip to chip communications



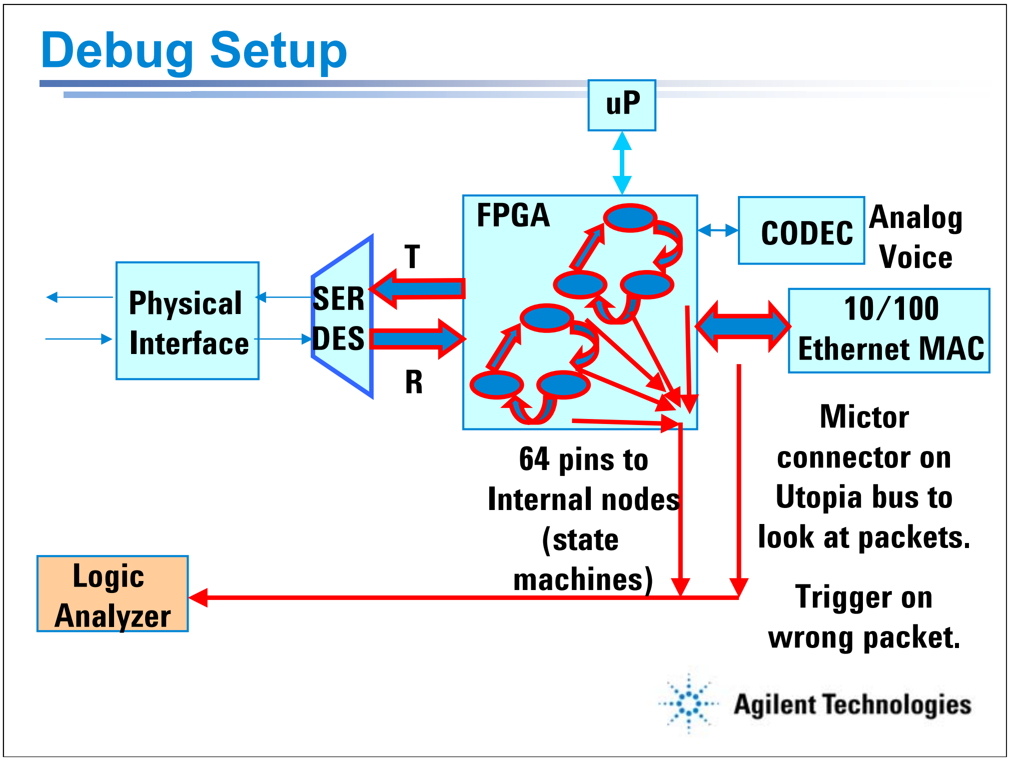
Debug Approach

- Chose large FPGA to prototype
- Prototyping board had other elements
- Dedicated 64 pins for debug and routed to Mictor connectors
- Put a Mictor connector on Utopia bus to view traffic

Agilent Technologies

In this situation, the designer chose a large FPGA, many times that needed for the final design, for use in developing the system on a prototyping board. This prototyping board had the things they anticipated for the final design, all the optical circuitry, a microprocessor, and Ethernet.

The strategy was to get Mictor connectors on the board from the get go, and they dedicated 64 pins on the FPGA for debug. He placed a Mictor connector on a utopia bus in order to view traffic.



Debug Setup

uP

FPGA

CODEC

Analog Voice

10/100 Ethernet MAC

Physical Interface

SER DES

T

R

64 pins to Internal nodes (state machines)

Mictor connector on Utopia bus to look at packets.

Trigger on wrong packet.

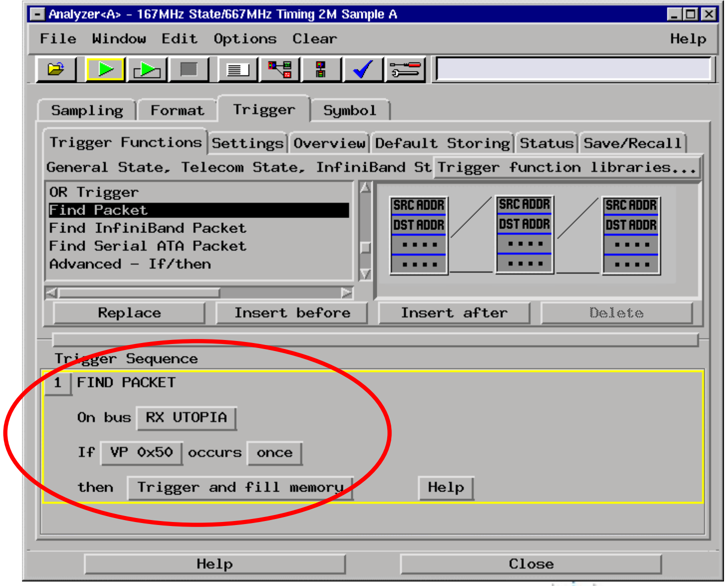
Logic Analyzer

Agilent Technologies

Key to the approach is triggering at a high packet level and then viewing the condition of multiple state machines inside the FPGA via numerous debug pins.

Their goal was to trigger the logic analyzer from the utopia bus, where they were seeing wrong packets, and then look at the exact condition of the state machines to try and sort out what was going wrong.

Packet Level Trigger


Analyzer<A> - 167MHz State/667MHz Timing 2M Sample A
File Window Edit Options Clear Help
Sampling Format Trigger Symbol
Trigger Functions Settings Overview Default Storing Status Save/Recall
General State, Telecom State, InfiniBand St Trigger function libraries...
OR Trigger
Find Packet
Find InfiniBand Packet
Find Serial ATA Packet
Advanced - If/then
SRC ADDR
DST ADDR
Replace Insert before Insert after Delete
Help Close

Trigger Sequence

1 FIND PACKET

On bus RX UTOPIA

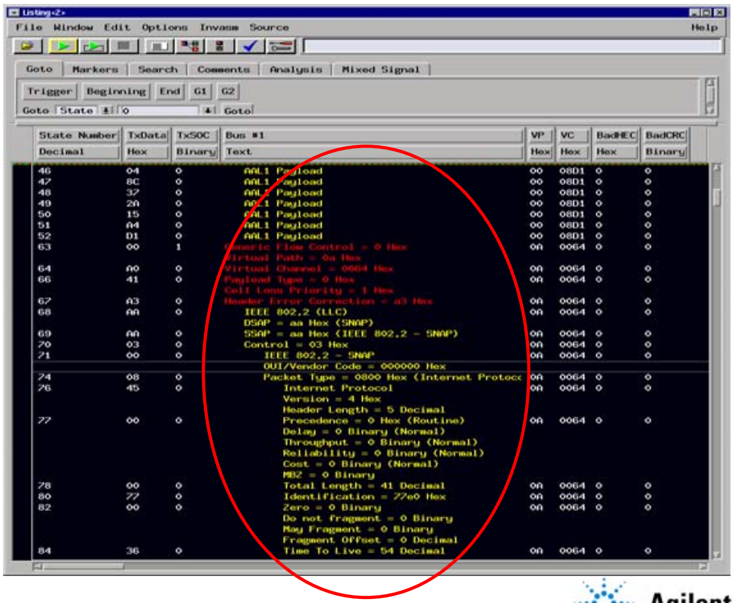
If VP 0x50 occurs once

then Trigger and fill memory Help

Agilent Technologies

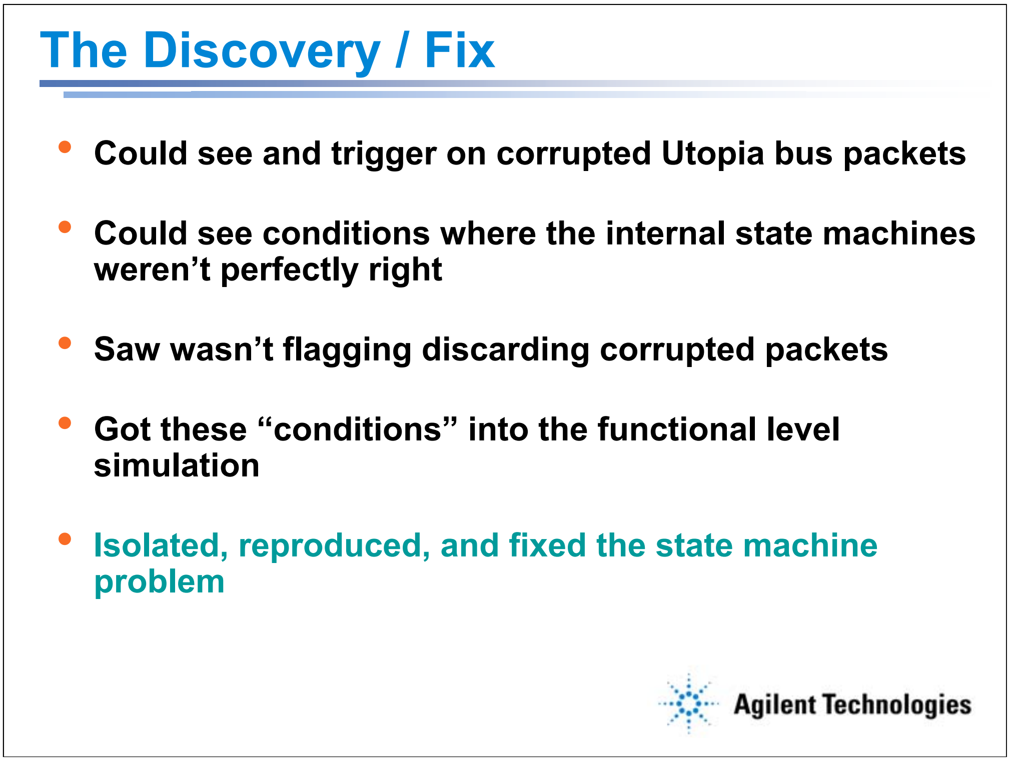
The logic analyzer allowed the designer to set up a specific Utopia bus packet condition to trigger on.

Protocol Decode View



```
State Number  TxData  TxSOC   Bus #1                                      VP   VC    BadHEC  BadCRC
Decimal       Hex     Binary  Text                                        Hex  Hex   Hex     Binary
46            04      0           AAL1 Payload                            00   08D1  0       0
47            8C      0           AAL1 Payload                            00   08D1  0       0
48            37      0           AAL1 Payload                            00   08D1  0       0
49            2A      0           AAL1 Payload                            00   08D1  0       0
50            15      0           AAL1 Payload                            00   08D1  0       0
51            A4      0           AAL1 Payload                            00   08D1  0       0
52            D1      0           AAL1 Payload                            00   08D1  0       0
63            00      1       Generic Flow Control = 0 Hex                0A   0064  0       0
                              Virtual Path = 0a Hex
64            A0      0       Virtual Channel = 0064 Hex                  0A   0064  0       0
66            41      0       Payload Type = 0 Hex                        0A   0064  0       0
                              Cell Loss Priority = 1 Hex
67            A3      0       Header Error Correction = a3 Hex            0A   0064  0       0
68            AA      0           IEEE 802.2 (LLC)                        0A   0064  0       0
                                  DSAP = aa Hex (SNAP)
69            AA      0           SSAP = aa Hex (IEEE 802.2 - SNAP)       0A   0064  0       0
70            03      0           Control = 03 Hex                        0A   0064  0       0
71            00      0             IEEE 802.2 - SNAP                     0A   0064  0       0
                                    OUI/Vendor Code = 000000 Hex
74            08      0             Packet Type = 0800 Hex (Internet Protoco 0A 0064  0       0
76            45      0               Internet Protocol                   0A   0064  0       0
                                      Version = 4 Hex
                                      Header Length = 5 Decimal
77            00      0               Precedence = 0 Hex (Routine)        0A   0064  0       0
                                      Delay = 0 Binary (Normal)
                                      Throughput = 0 Binary (Normal)
                                      Reliability = 0 Binary (Normal)
                                      Cost = 0 Binary (Normal)
                                      MBZ = 0 Binary
78            00      0               Total Length = 41 Decimal           0A   0064  0       0
80            77      0               Identification = 77e0 Hex           0A   0064  0       0
82            00      0               Zero = 0 Binary                     0A   0064  0       0
                                      Do not fragment = 0 Binary
                                      May Fragment = 0 Binary
                                      Fragment Offset = 0 Decimal
84            36      0               Time To Live = 54 Decimal           0A   0064  0       0
```

Now it was possible to see the traffic on the Utopia bus, not just in low level Hex values, but also at a protocol level.



The Discovery / Fix

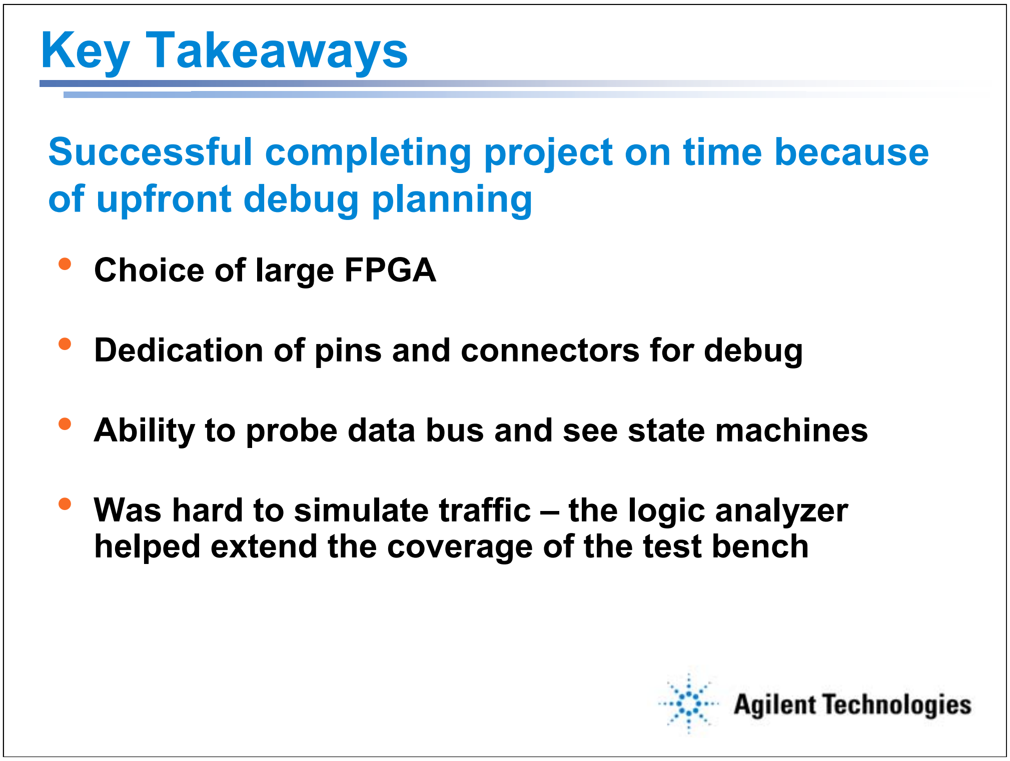
- Could see and trigger on corrupted Utopia bus packets
- Could see conditions where the internal state machines weren’t perfectly right
- Saw wasn’t flagging discarding corrupted packets
- Got these “conditions” into the functional level simulation
- Isolated, reproduced, and fixed the state machine problem

Agilent Technologies

He watched the utopia bus at a packet level with the Datacom Tool Set in the logic analyzer. He was able to see and trigger on corrupted packets as hoped.

As he looked at the state machine activity inside the FPGA, immediately before the corrupted packet, he could seek the exact condition where one of the state machines wasn't perfectly right. In fact, corrupted packets were supposed to be detected and flagged as something to be discarded, but that wasn't happening. So it was normal that the memory might overflow in heavy traffic situations and packets get corrupted, but it was unacceptable to fail to flag the errors.

Key to the process was getting these conditions into the functional level simulation. Then there were able to isolate, reproduce in simulation, and then fix in simulation, the state machine problem.



Key Takeaways

Successful completing project on time because of upfront debug planning

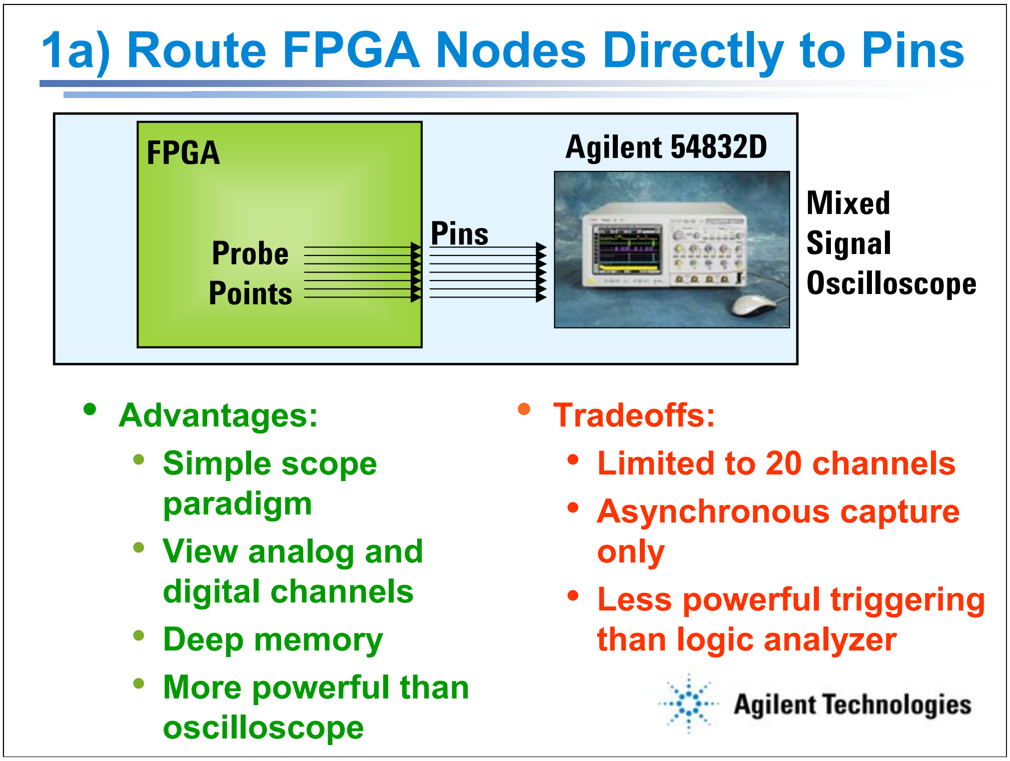
- Choice of large FPGA
- Dedication of pins and connectors for debug
- Ability to probe data bus and see state machines
- Was hard to simulate traffic – the logic analyzer helped extend the coverage of the test bench

Agilent Technologies

A few takeaways from this example:

The engineer really valued having a lot of debug pins and chose the FPGA accordingly.

It was critical of First probe the data bus and then see the condition of the internal state machines. The engineer pointed out that it was a very difficult to stimulate traffic, and by identifying specific traffic conditions that caused problems, he was able to extend the coverage of the test bench in simulation.



1a) Route FPGA Nodes Directly to Pins

- Advantages:
  - Simple scope paradigm
  - View analog and digital channels
  - Deep memory
  - More powerful than oscilloscope
- Tradeoffs:
  - Limited to 20 channels
  - Asynchronous capture only
  - Less powerful triggering than logic analyzer

Agilent Technologies

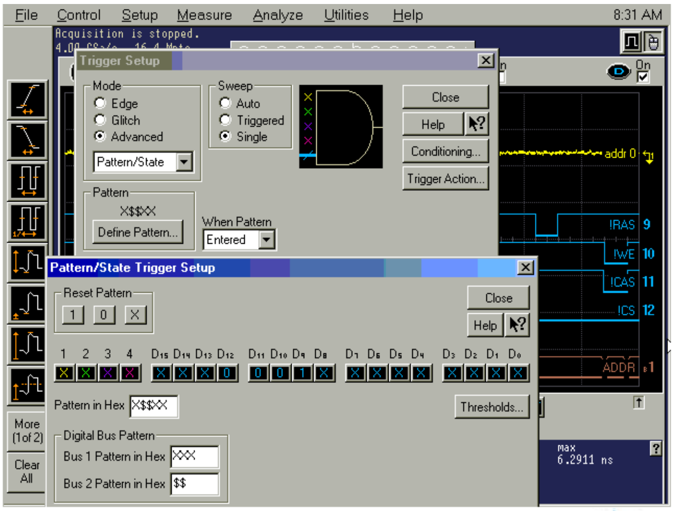
One variation to this first approach is to use a Mixed Signal Oscilloscope, or MSO to monitor signals on FPGA debug pins or other pins of the FPGA instead of using a logic analyzer.

An MSO can serve as a basic, easy to use tool for observing FPGA activity as well as observing signals on the FPGA boundary as the FPGA interfaces to other parts of a system.   Because an MSO offers 4 analog channels plus another 16 digital channels with deep memory capture, there's the opportunity to observe operations like state machine activity, with some hope of triggering on a symptom of a problem, while looking backward in time to find a root cause of a problem. 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.

With the analog channels of an MSO or a standard oscilloscope, one has high resolution to view signal characteristics at the FPGA boundary. Such signals must meet specifications to interface to the rest of the system. An oscilloscope can make detailed timing measurements and view signal integrity characteristics of signals present. Some of these scope capabilities are less pertinent if trying to observe activity "inside" the FPGA via debug pins. Because of the routing delays to debug pins, channel to channel skew is introduced which obscures the actual timing present inside the FPGA

Mixed Signal Oscilloscope Setup

Define digital labels, define bus, define trigger



Acquisition is stopped.

Trigger Setup

Mode: ○ Edge ○ Glitch ● Advanced — Pattern/State

Sweep: ○ Auto ○ Triggered ● Single

Pattern: X$$XX

When Pattern: Entered

Pattern/State Trigger Setup

Reset Pattern: 1 0 X

1	2	3	4	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
X	X	X	X	X	X	X	0	0	0	1	X	X	X	X	X	X	X	X	X

Pattern in Hex: X$$XX

Digital Bus Pattern
Bus 1 Pattern in Hex: XXX
Bus 2 Pattern in Hex: $$

Agilent Technologies

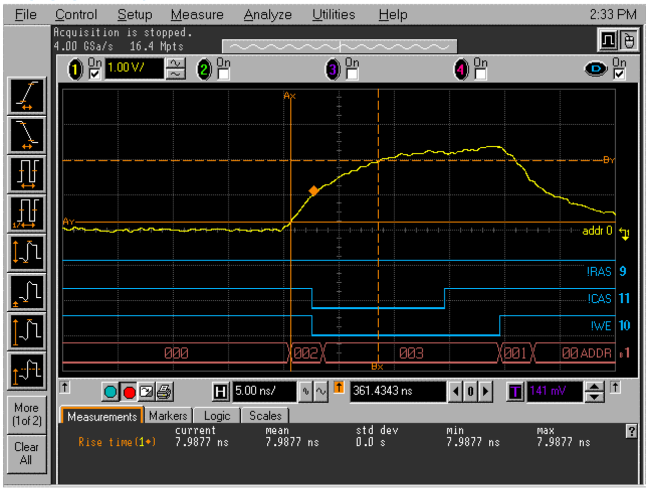
In many digital applications today, it has become important to be able to look at digital signals of interest under very specific target conditions, such as a memory write in SDRAM. Traditional oscilloscopes, with only 2 or 4 analog input channels, aren't able to trigger on the more complex control lines that define such conditions.

Now, with Mixed Signal Oscilloscope, it's very simple to label bits of interest, including multi-signal buses, and then define a trigger condition of interest.

Here, the MSO triggers when RAS is high, CAS low, Write Enable low, Chip Select low, and on the rising edge of clock.

Identify Slow Edge on Address Bit

Trigger on and View SDRAM Control Signals, Address Bit

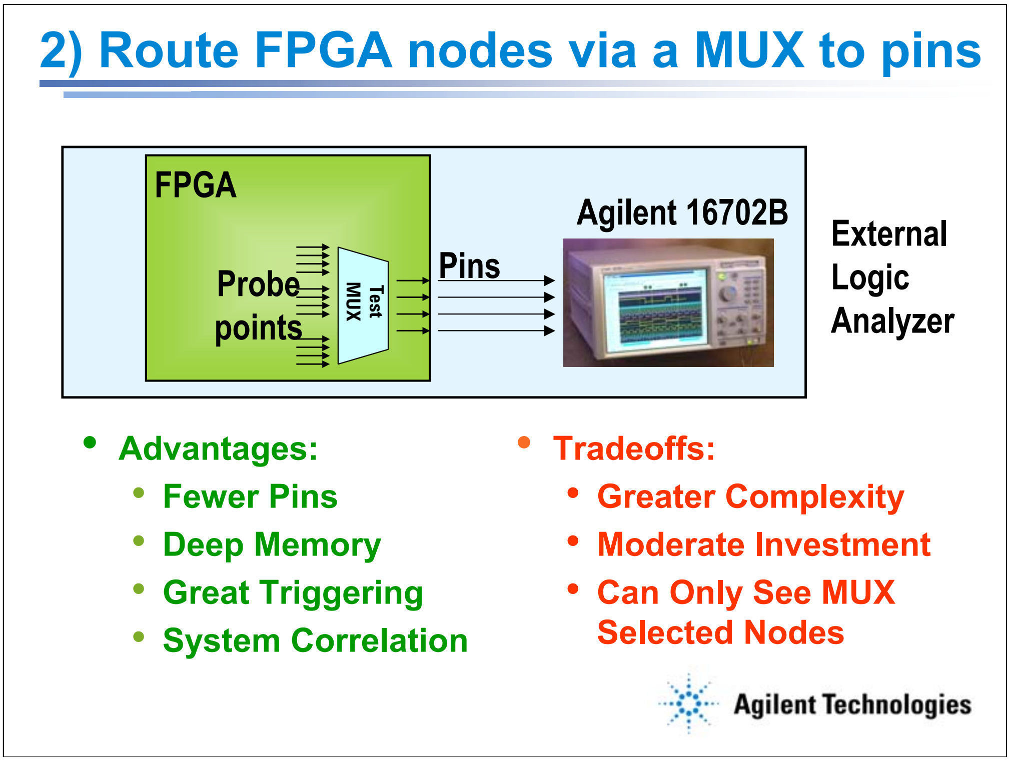


```
Acquisition is stopped.
4.00 GSa/s   16.4 Mpts
```

```
                current     mean        std dev     min         max
Rise time(1↑)   7.9877 ns   7.9877 ns   0.0 s       7.9877 ns   7.9877 ns
```

Agilent Technologies

Notice that the address does not cleanly change from 000 to 003 and back, but instead takes a while to finally reach a 003 value. Having triggered on a memory write, and then scrolled through the trace to this problem point, an analog channel was then used to view the problem bit to look for the root cause of the problem. It wasn't skew or ground bounce, but rather, it was a slow edge, perhaps from a weak driver.



2) Route FPGA nodes via a MUX to pins

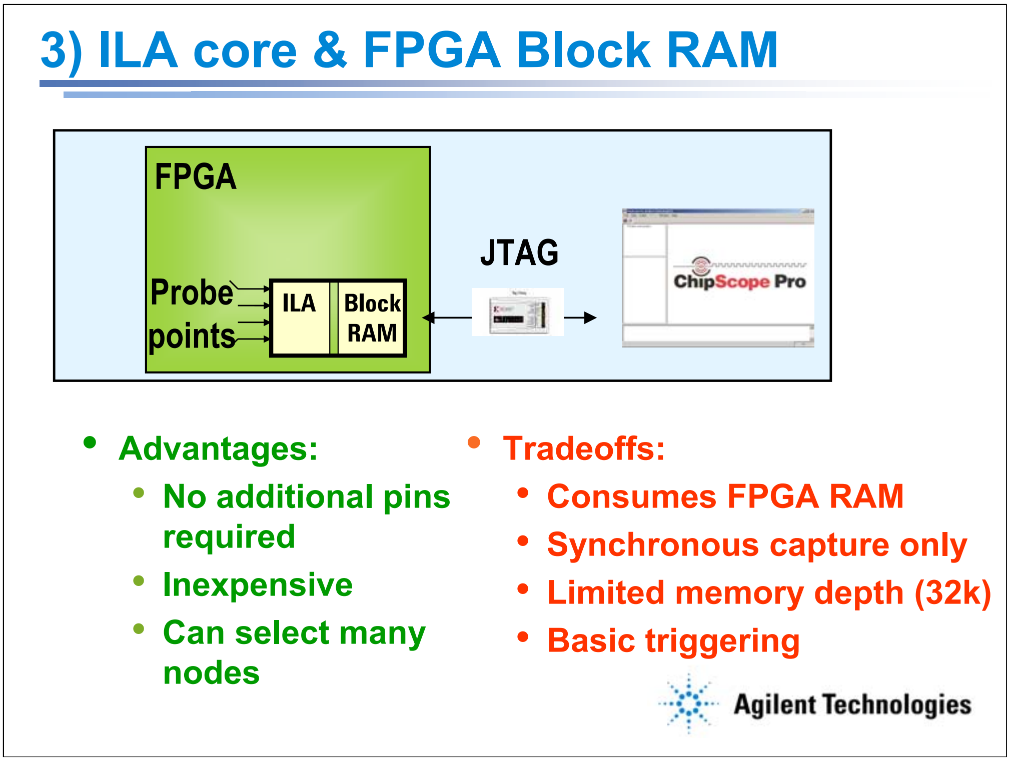
- Advantages:
  - Fewer Pins
  - Deep Memory
  - Great Triggering
  - System Correlation
- Tradeoffs:
  - Greater Complexity
  - Moderate Investment
  - Can Only See MUX Selected Nodes

Agilent Technologies

The second approach is similar to the first except that internal nodes are routed to a MUX and fewer pins are dedicated to debug. The disadvantage here is managing the MUX process and you can’t see all the signals at once.

Our research shows that many designers build in a test MUX that allows a customer to select a single group of signals to bring out to the pins at a single time. This technique minimizes the number of pins dedicated for debug. Typcial pin count for debug is in the range of 8 to 32 pins while internal nodes feeding the MUX is typically in the order of 32 to a hundred channels. When the customer finishes debug, the text MUX circuitry and pins for debug remain in the design. Taking them out would change the design characteristics and require a new round of debug.

This approach also has the variation of using an MSO or traditional scope to observe signals.



3) ILA core & FPGA Block RAM

- Advantages:
  - No additional pins required
  - Inexpensive
  - Can select many nodes
- Tradeoffs:
  - Consumes FPGA RAM
  - Synchronous capture only
  - Limited memory depth (32k)
  - Basic triggering

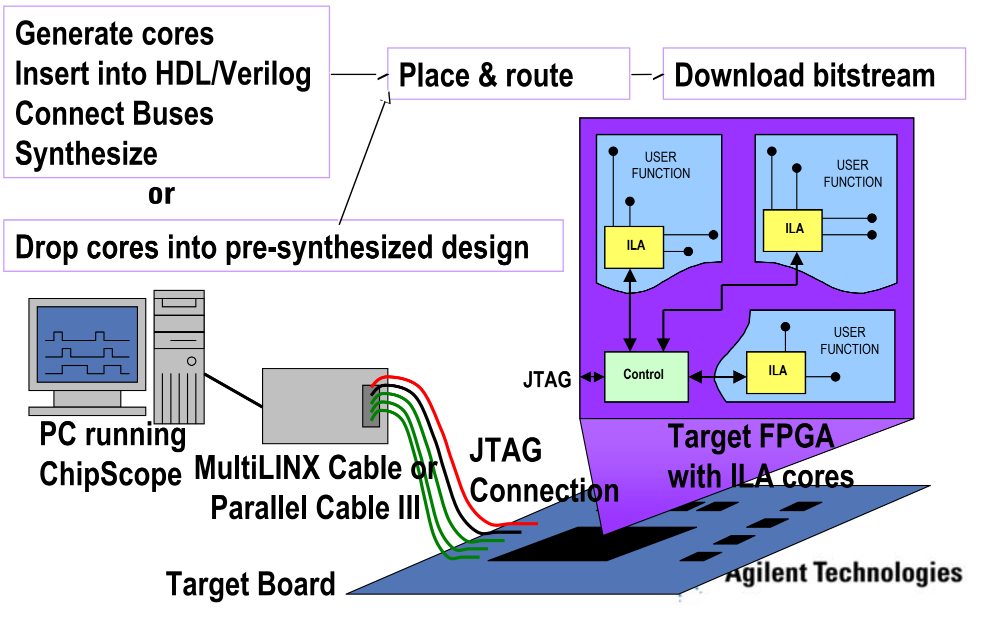
Agilent Technologies

Several vendors offer options to embed logic analyzer cores inside their FPGAs. For example, Xilinx has Integrated Logic Analyzer (ILA) cores including internal block RAM to store signals. Nodes of interest are selected, traces captured, and then output via JTAG to their ChipScope software.

One big advantage here is that no FPGA pins are required beyond the JTAG connection. This is particularly useful for designs that do not have pins available for debug. It's also an inexpensive solution.

There are a number of tradeoffs. These include limited memory, the consumption of important FPGA memory, and limited triggering. As with all cores, inclusion of a logic analyzer core will have an impact on the design itself. For this reason, many engineers prefer to leave the core in the FPGA even after the design is debugged.

Options to Insert Cores



Generate cores
Insert into HDL/Verilog
Connect Buses
Synthesize

or

Drop cores into pre-synthesized design

Place & route

Download bitstream

PC running ChipScope

MultiLINX Cable or Parallel Cable III

JTAG Connection

Target FPGA with ILA cores

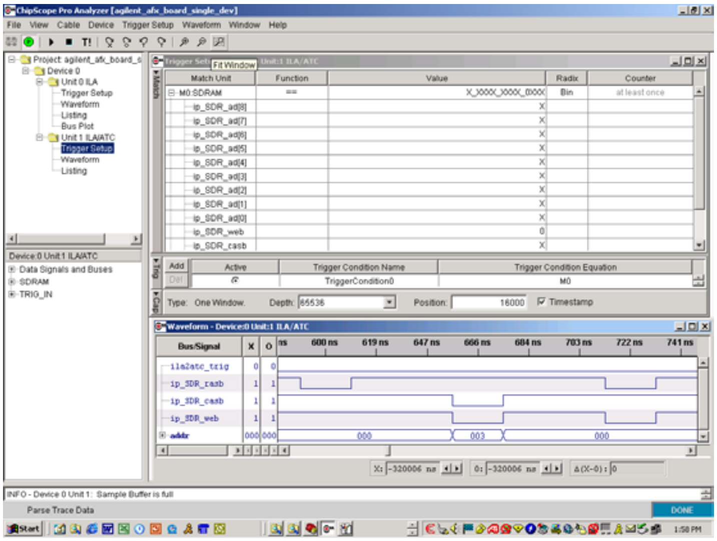
Target Board

There are a couple of ways to insert ILA blocks into a design. The first involves inserting them at the verilog level synthesizing the design and then doing a place-and-route. At that point, the ChipScope software can download a bitstream to the FPGA to program it with a design including the ILA blocks.

The other approach is to take a presynthesized design, drop ILA cores into that design, then place-and-route, and download the design into the FPGA.

You’ll see there is actually a JTAG control block in the FPGA that interfaces with the logic analyzer blocks. Data is pulled out by this control block through the JTAG, back to the PC, and then presented on the ChipScope interface.

ChipScope Pro Analyzer View



Match Unit	Function	Value	Radix	Counter
M0:SDRAM	==	X_XXXX_XXXX_0XXX	Bin	at least once
ip_SDR_ad[8]		X		
ip_SDR_ad[7]		X		
ip_SDR_ad[6]		X		
ip_SDR_ad[5]		X		
ip_SDR_ad[4]		X		
ip_SDR_ad[3]		X		
ip_SDR_ad[2]		X		
ip_SDR_ad[1]		X		
ip_SDR_ad[0]		X		
ip_SDR_web		0		
ip_SDR_casb		X		

Add / Del	Active	Trigger Condition Name	Trigger Condition Equation
	●	TriggerCondition0	M0

Type: One Window. Depth: 65536 Position: 16000 ☑ Timestamp

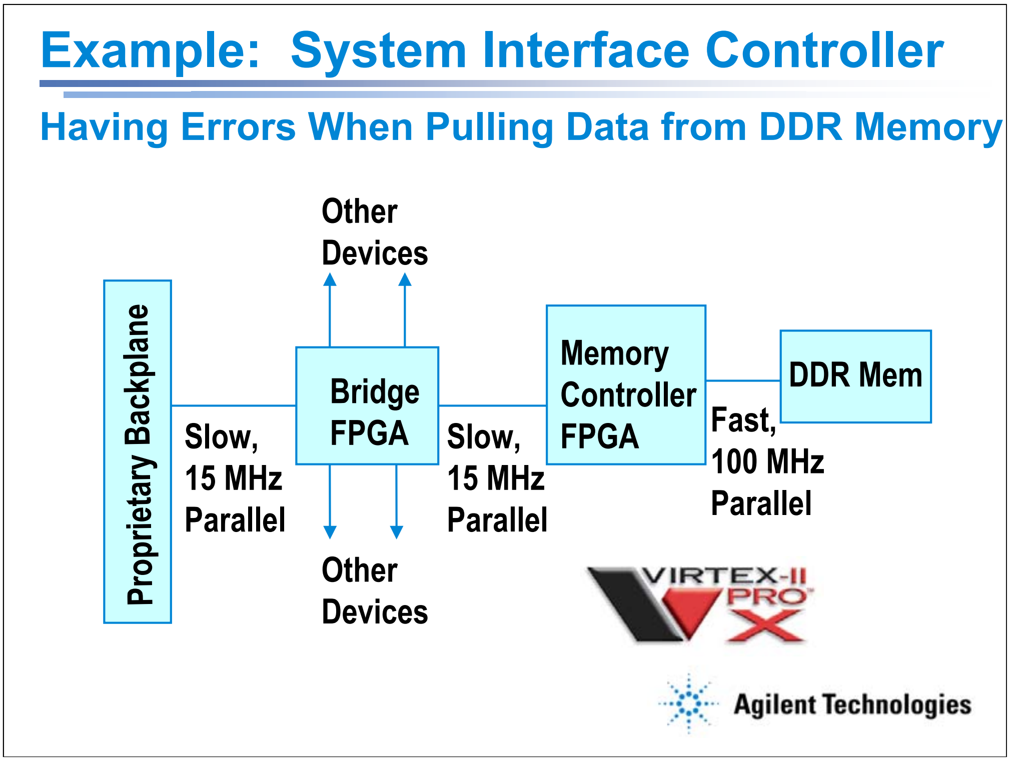
Bus/Signal	X	O
ila2atc_trig	0	0
ip_SDR_rasb	1	1
ip_SDR_casb	1	1
ip_SDR_web	1	1
addr	000	000

X: -320006 ns O: -320006 ns Δ(X-O): 0

INFO - Device 0 Unit 1: Sample Buffer is full

Parse Trace Data

Here's a view of the ChipScope Pro interface in its logic analyzer screen.



Example: System Interface Controller

Having Errors When Pulling Data from DDR Memory

Proprietary Backplane

Slow, 15 MHz Parallel

Bridge FPGA

Other Devices

Other Devices

Slow, 15 MHz Parallel

Memory Controller FPGA

Fast, 100 MHz Parallel

DDR Mem

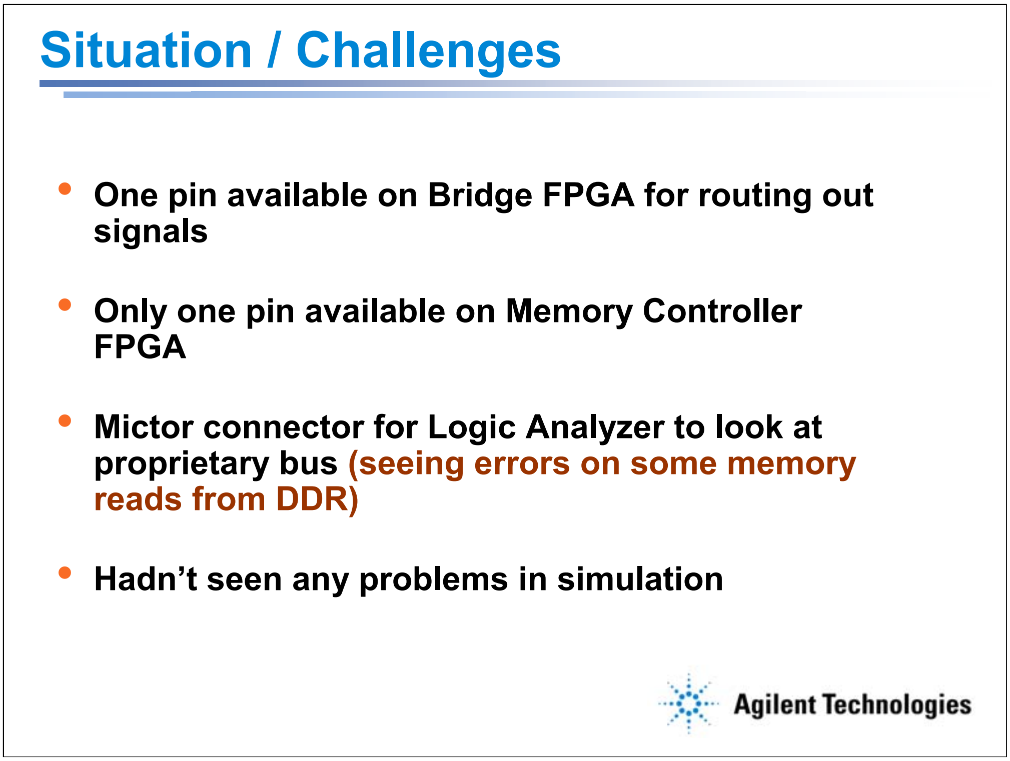
VIRTEX-II PRO X

Agilent Technologies

One engineering example involves a case where a system interface controller was experiencing errors when a 15 MHz proprietary backplane tried to read DDR memory via a couple of FPGAs.

The first FPGA really served as a bridge allowing the selection of multiple devices, one being the DDR memory. The second FPGA served as a DDR memory controller and interface between the 100 MHz DDR and the 15 MHz system bus.

Both FPGAs were from the Xilinx Virtex II family.



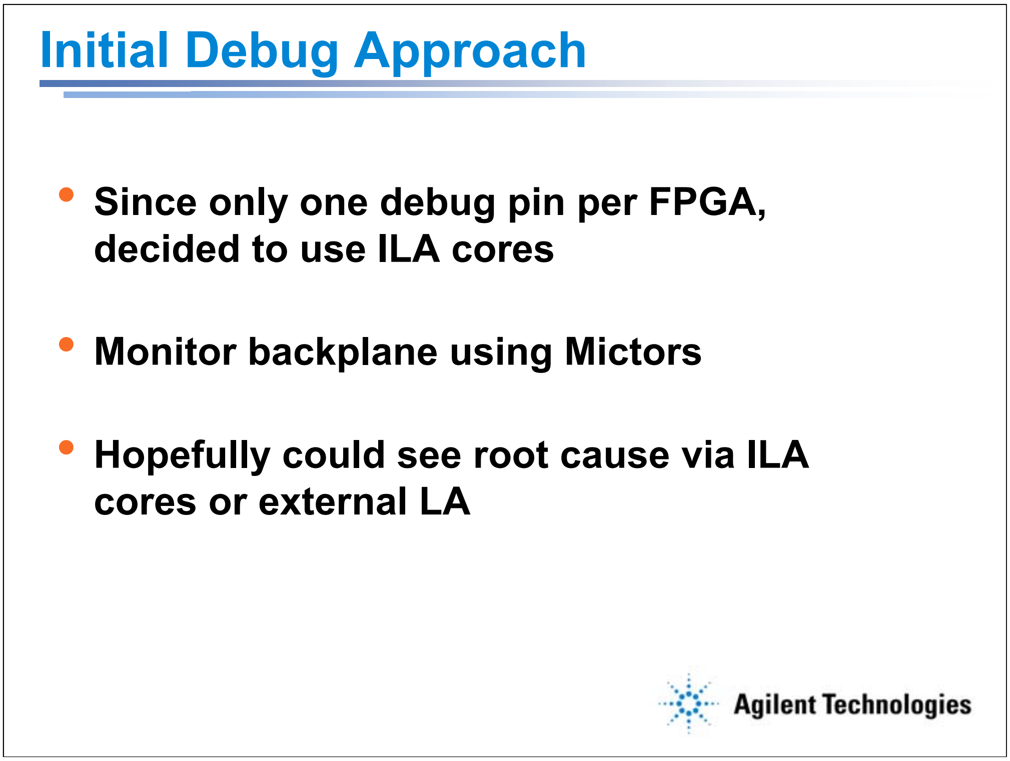
Situation / Challenges

- One pin available on Bridge FPGA for routing out signals
- Only one pin available on Memory Controller FPGA
- Mictor connector for Logic Analyzer to look at proprietary bus (seeing errors on some memory reads from DDR)
- Hadn’t seen any problems in simulation

Agilent Technologies

The designer had some tough constraints to work with. First there was only one pin available on the bridge FPGA for routing out internal nodes, and only one pin on the memory controller FPGA. The designer had placed a Mictor connector on its board to be able to look at a proprietary bus with the logic analyzer.

There were errors on some memory reads from DDR, but no errors had been seen in simulation

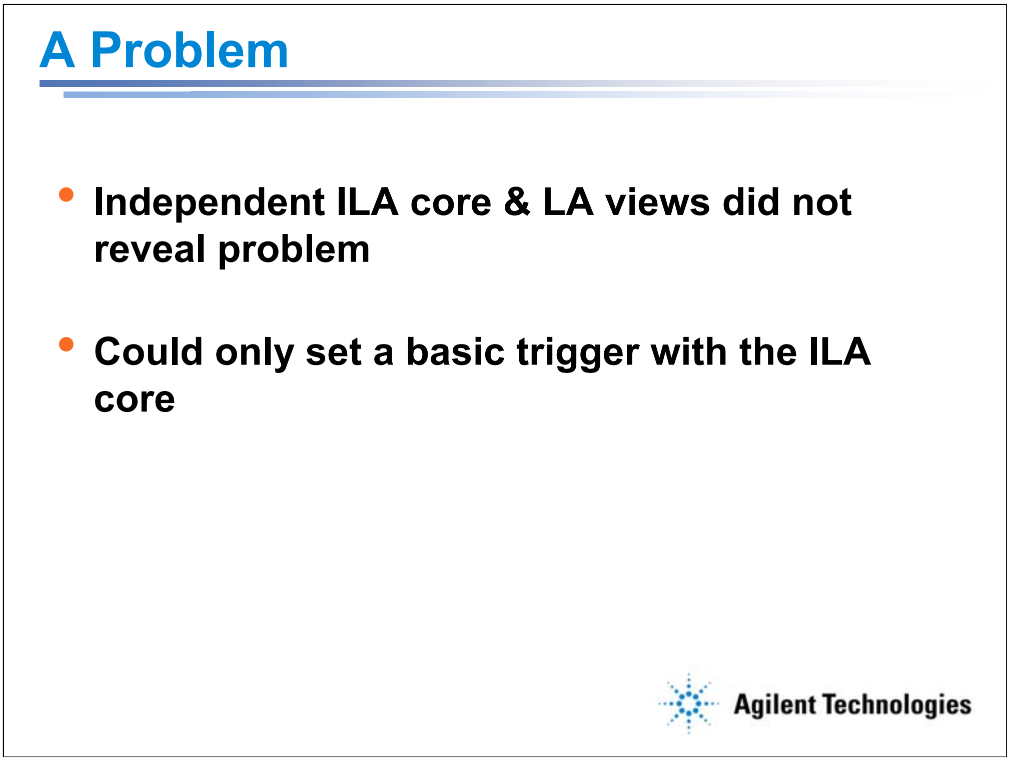


Initial Debug Approach

- Since only one debug pin per FPGA, decided to use ILA cores
- Monitor backplane using Mictors
- Hopefully could see root cause via ILA cores or external LA

Agilent Technologies

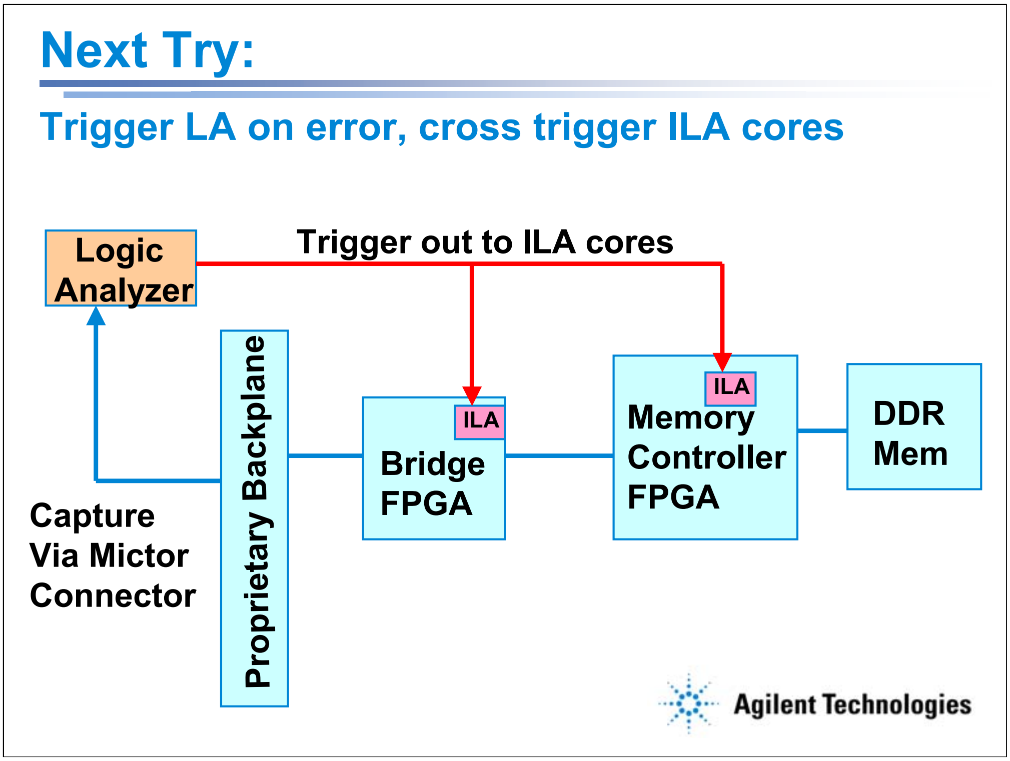
Given that there was only one FPGA debug pin per FPGA the designer felts he would need to use integrated logic analyzer cores in the FPGAs. He planned to monitor the back plane with logic analyzer, and his hope was that he'd see the source of the problem either by looking at the internal FPGA activity or the external bus activity.



A Problem

- Independent ILA core & LA views did not reveal problem
- Could only set a basic trigger with the ILA core

That was hopeful thinking, but the independent ILA core and logic analyzer views did not reveal what the root cause of the problem was. One problem was that the ILA core approach could only have a simple trigger set for them.



Next Try:

Trigger LA on error, cross trigger ILA cores

Logic Analyzer

Trigger out to ILA cores

Proprietary Backplane

ILA

Bridge FPGA

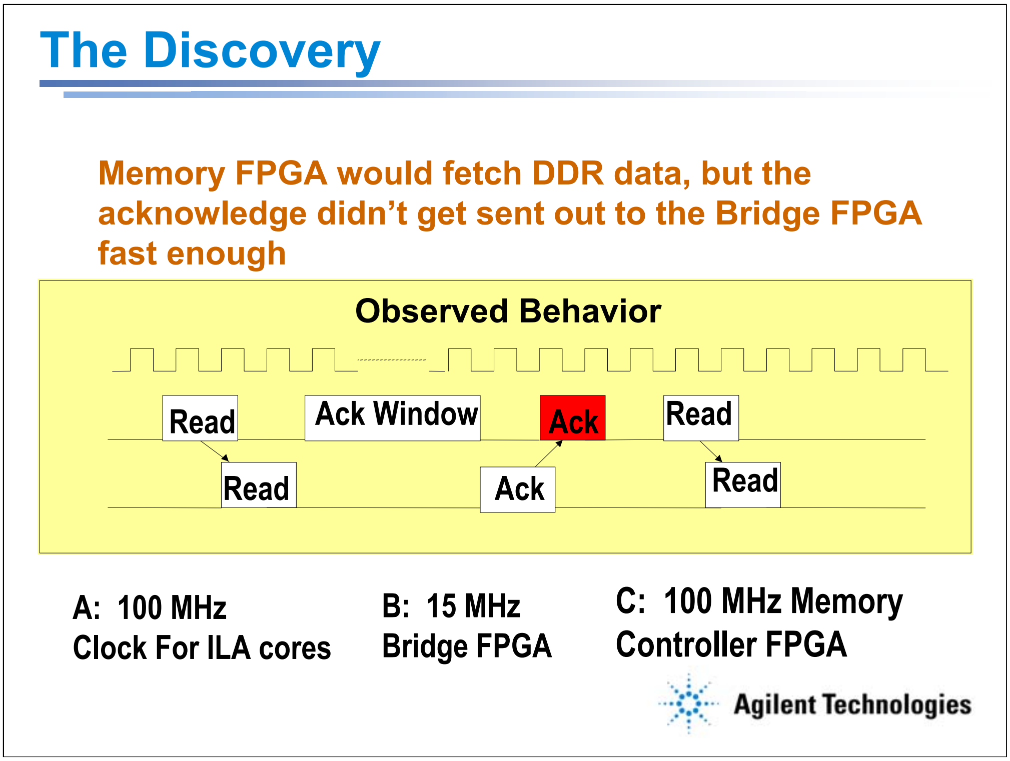
ILA

Memory Controller FPGA

DDR Mem

Capture Via Mictor Connector

So this was going to take a more complex approach for debug. The designer decided to trigger the logic analyzer when he saw an error on the proprietary back plane, but then trigger the integrated logic analyzer blocks on the inside each of the FPGA case. The goal was to try and understand exactly what was happening inside the FPGAs at the point when the error occurred, and to possibly have to look back in time for the sequence of steps that led up to the visible error seen out on the back plane. Part of the challenge was that it was a multiplexed Address/Data bus on the backplane, and a logic analyzer was necessary to de-multiplex that bus so that you could then trigger on it.



The Discovery

Memory FPGA would fetch DDR data, but the acknowledge didn't get sent out to the Bridge FPGA fast enough

Observed Behavior

Read

Ack Window

Ack

Read

Read

Ack

Read

A: 100 MHz
Clock For ILA cores

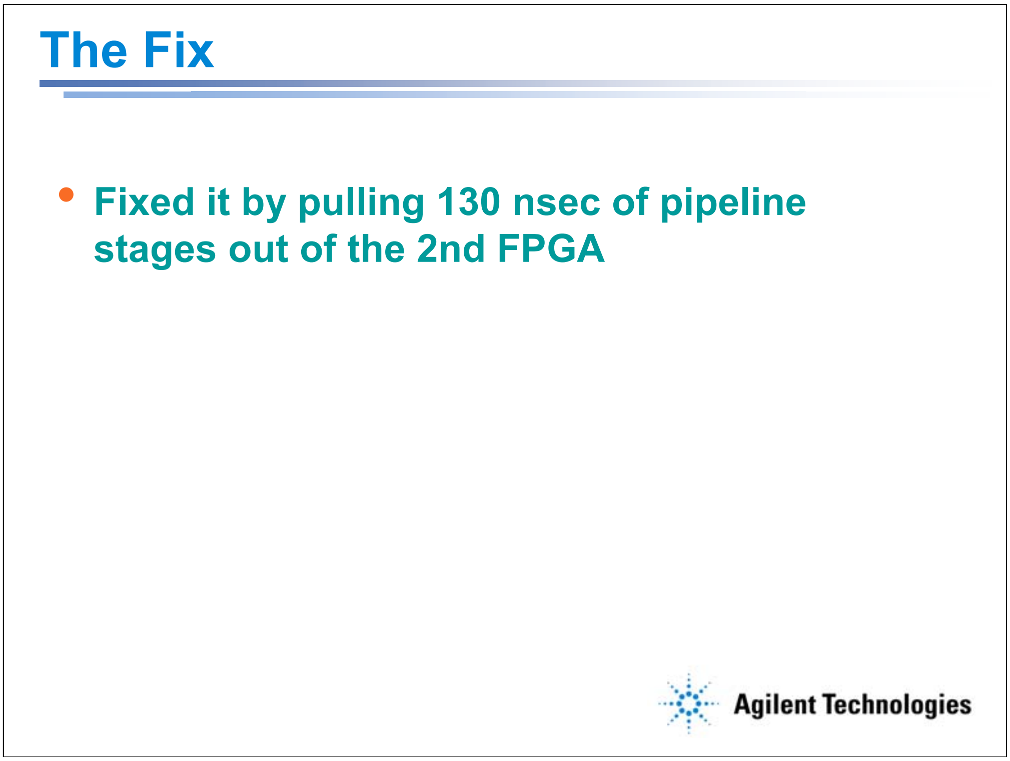
B: 15 MHz
Bridge FPGA

C: 100 MHz Memory
Controller FPGA

Agilent Technologies

Well that was the right approach. Cross triggering from the back plane condition to the logic analyzer blocks inside the FPGA did in fact uncovered the root cause of the failure. What was happening was when the bridge FPGA requested a read, the memory controller FPGA did a read from 100 MHz DDR memory. But then the 15 MHz bridge FPGA needed to receive an acknowledge signal within a certain time. By looking inside the FPGAs, it became obvious that the acknowledge was coming outside that time window. So this was a timing error, but it didn't deal with set up and hold types of timing problems surrounding the clock edge. This was a timing error that dealt with a sequencing of events between two time domains.

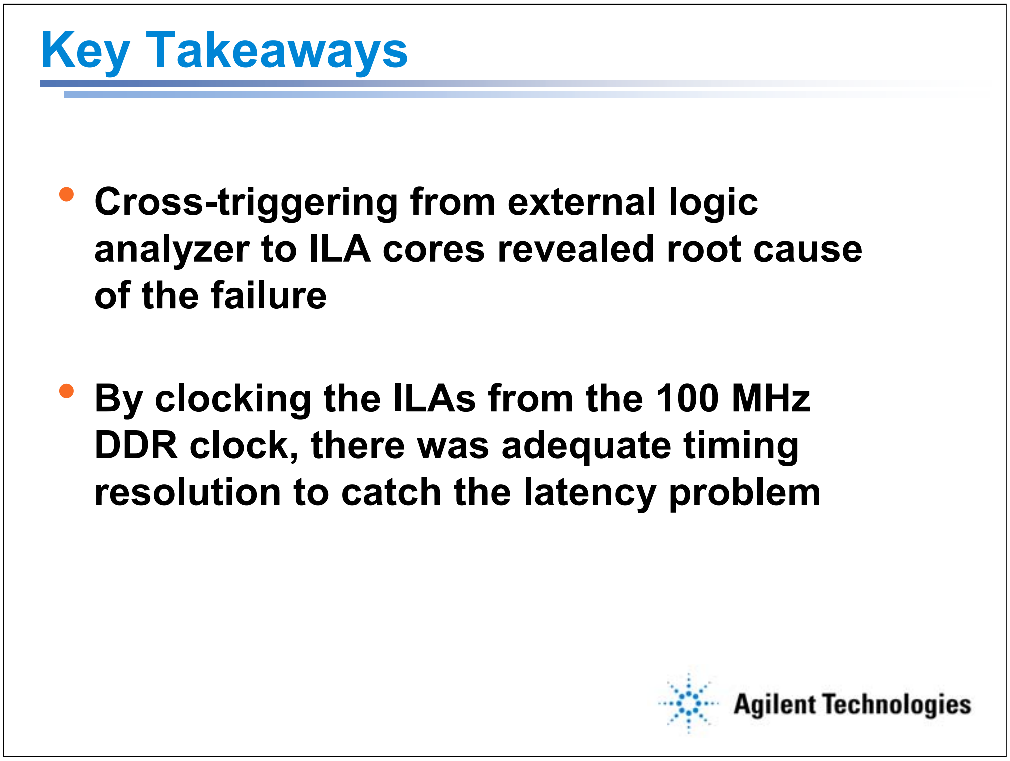
The designer saw the acknowledge problem with the ILA block inside the Bridge FPGA. When looking just at the memory controller FPGA initially without the cross-trigger from the logic analyzer, he could see good data was making it out of the memory FPGA. It was only with a cross trigger, and an ability to see inside the bridge FPGA at the precise time that error occurred on the backplane, that he could catch the acknowledge timing problem.



The Fix

- Fixed it by pulling 130 nsec of pipeline stages out of the 2nd FPGA

So the designer had to pull 130 nanoseconds of pipeline stages out of the second FPGA in order to pull this acknowledge signal back in.



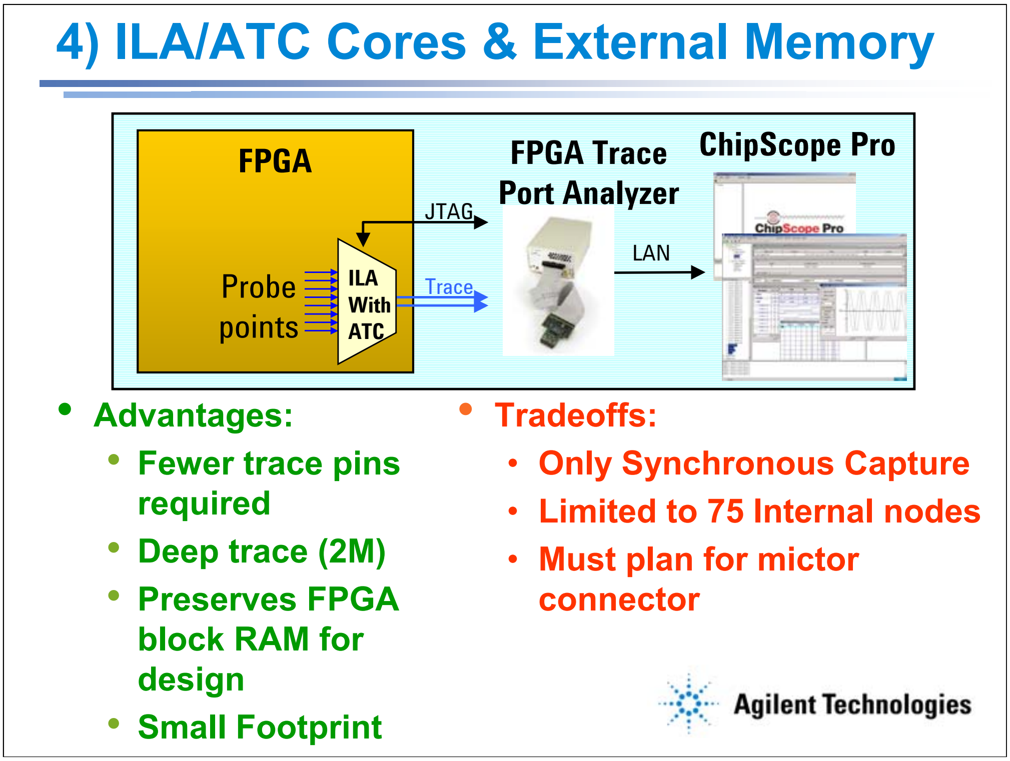
Key Takeaways

- Cross-triggering from external logic analyzer to ILA cores revealed root cause of the failure
- By clocking the ILAs from the 100 MHz DDR clock, there was adequate timing resolution to catch the latency problem

Agilent Technologies

It was important to be able to cross trigger from a symptom of the problem with an external logic analyzer, to the ILA blocks inside the FPGAs, in order to see the root cause of the failure.

In this example, by clocking the ILAs from the 100 MHz DDR clock, there was good timing resolution to see the latency problem on the 15 MHz FPGA. ChipScope ILAs D. if asynchronous capture, but it almost served as a timing analyzer on the slower bus.



4) ILA/ATC Cores & External Memory

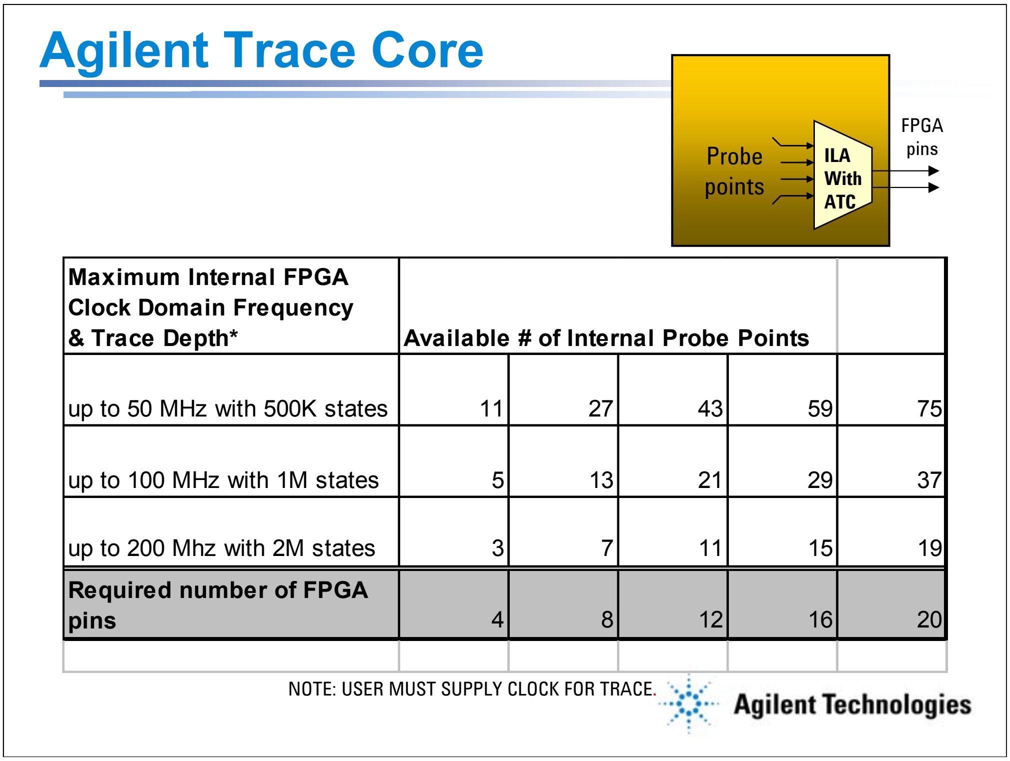
- Advantages:
  - Fewer trace pins required
  - Deep trace (2M)
  - Preserves FPGA block RAM for design
  - Small Footprint
- Tradeoffs:
  - Only Synchronous Capture
  - Limited to 75 Internal nodes
  - Must plan for mictor connector

Agilent Technologies

The fourth way of viewing internal nodes in an FPGA is a solution that involves new ILA cores Xilinx coupled with an Agilent trace core (ATC) and an Agilent FPGA Trace Port Analyzer (TPA).

Here, internal nodes are time division multiplexed out to external memory in the TPA. This approach has a number of advantages. The first can be fewer pins as well as up to 2-Meg deep memory, and that memory doesn’t draw on FPGA resources. Some of the tradeoffs are similar to the those we saw with the basic ILA approach.

Let’s look more closely at the ILA /ATC option.



Agilent Trace Core

Probe
points
ILA
With
ATC
FPGA
pins

Maximum Internal FPGA Clock Domain Frequency & Trace Depth*	Available # of Internal Probe Points				
up to 50 MHz with 500K states	11	27	43	59	75
up to 100 MHz with 1M states	5	13	21	29	37
up to 200 Mhz with 2M states	3	7	11	15	19
Required number of FPGA pins	4	8	12	16	20

NOTE: USER MUST SUPPLY CLOCK FOR TRACE.

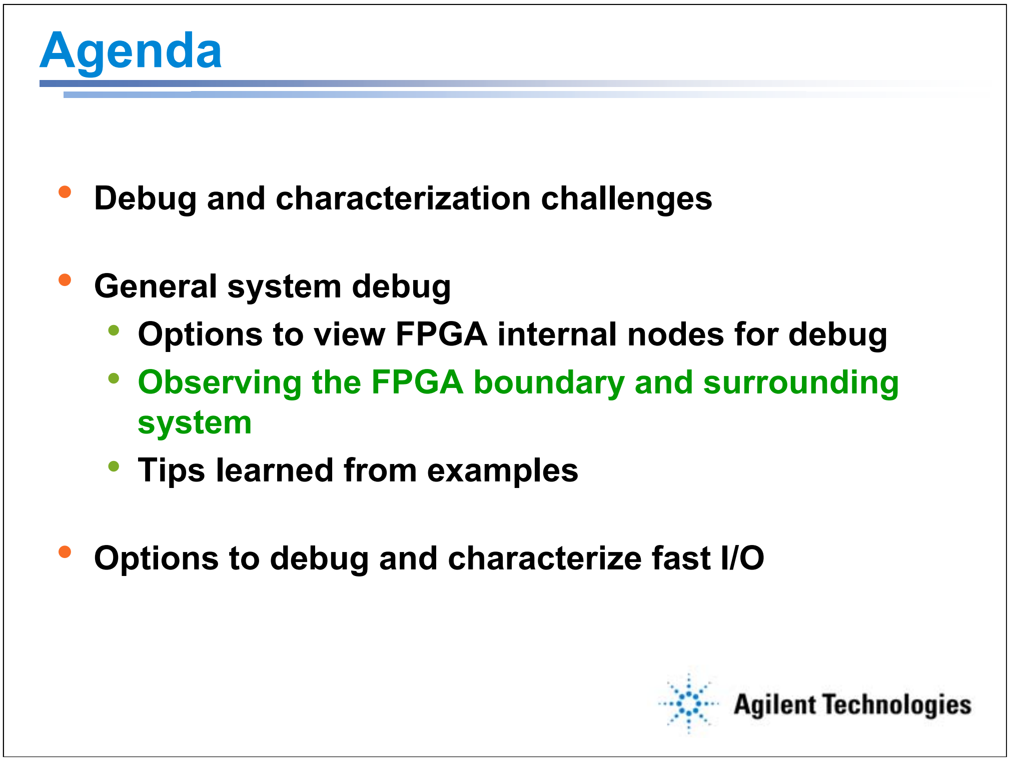
Agilent Technologies

Time division multiplexing via the Agilent Trace Core can be very advantageous to reduce pin count. For example, 75 signals running at 50 MHz can be 4:1 TDM'd down to only 20 pins while still giving full visibility to those signals.

Some details follow:

ATC comes in multiple pre-defined configurations depending on the clock speed of the internal circuit to be measured. ATC's TDM is based on the assumption that most FPGA circuits run at a fraction of the speed that the I/O pins are capable of (200 MHz single-ended).

For circuits up to 50 Mhz, this allows Agilent to use a 4:1 time-division MUX and monitor up to 75 probe points simultaneously at a memory depth of 256K states(samples). One channel is used for the clock and as MUXing ratios increase, some of the TPA storage resources are used for sync pulses so when the data is diplayed, triggering and packet information can be unraveled by ChipScope. For circuits running at >50 MHz to 100 MHz, one can use a 2:1 MUX.



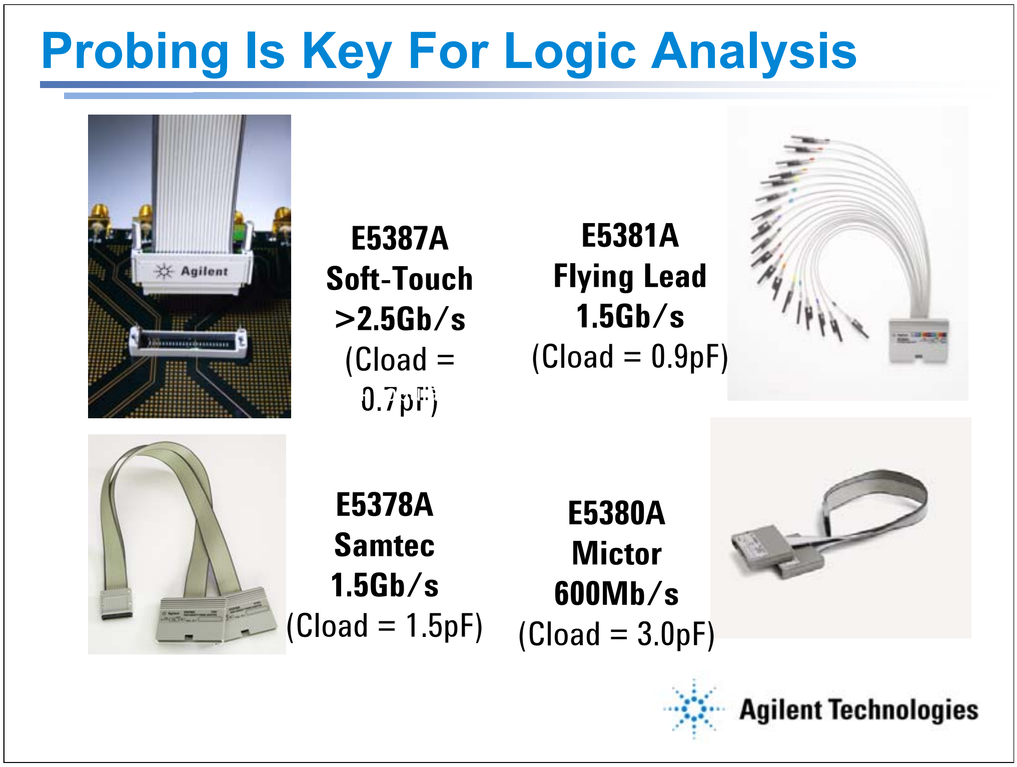
Agenda

- Debug and characterization challenges
- General system debug
  - Options to view FPGA internal nodes for debug
  - Observing the FPGA boundary and surrounding system
  - Tips learned from examples
- Options to debug and characterize fast I/O

Agilent Technologies

Agenda

So it's been important to look inside FPGAs for general debug, but it's also very important to look at the FPGA boundary and at other locations in a system at the board level.



Probing Is Key For Logic Analysis

E5387A
Soft-Touch
>2.5Gb/s
(Cload =
0.7pF)

E5381A
Flying Lead
1.5Gb/s
(Cload = 0.9pF)

E5378A
Samtec
1.5Gb/s
(Cload = 1.5pF)

E5380A
Mictor
600Mb/s
(Cload = 3.0pF)

Agilent
Agilent Technologies

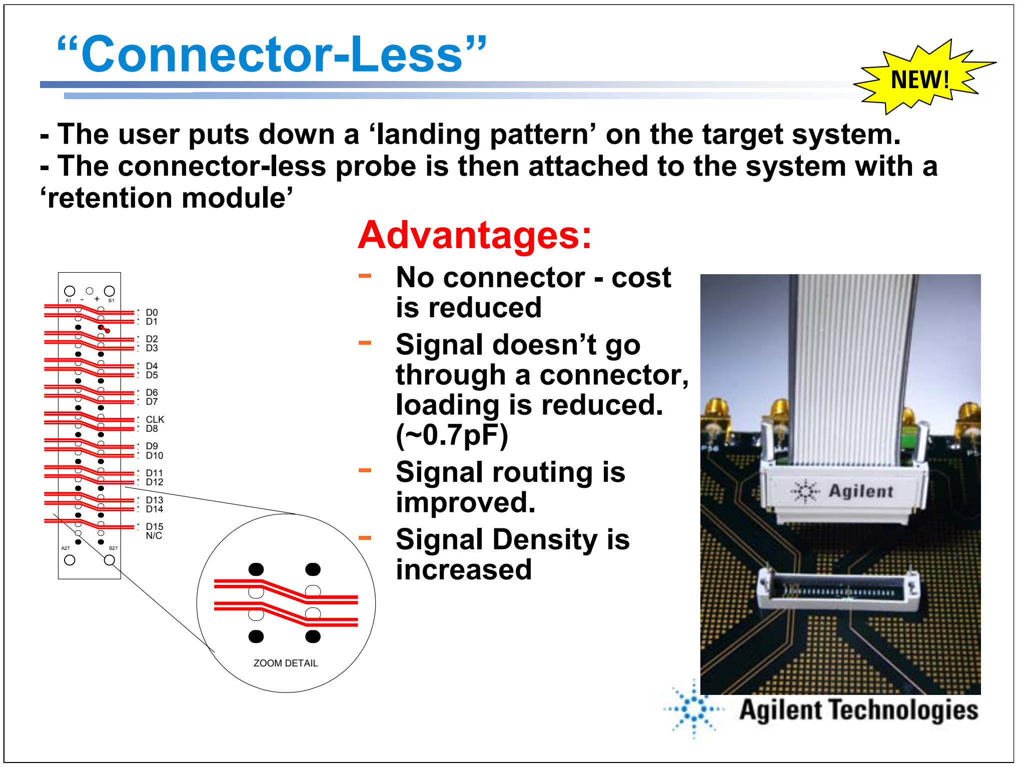
When trying to look at FPGA boundary signals or other signals in an overall system, planning for connecting in-circuit debug tools is important. In the last months, the options for such connections have greatly increased.

The evolution of Logic Analyzer Probes has given the user the following

-Lower Capacitive Loading

-Higher Resonant Frequency of the Probe Load

-Higher Bandwidth Probes



“Connector-Less”

NEW!

- The user puts down a ‘landing pattern’ on the target system.
- The connector-less probe is then attached to the system with a ‘retention module’

Advantages:

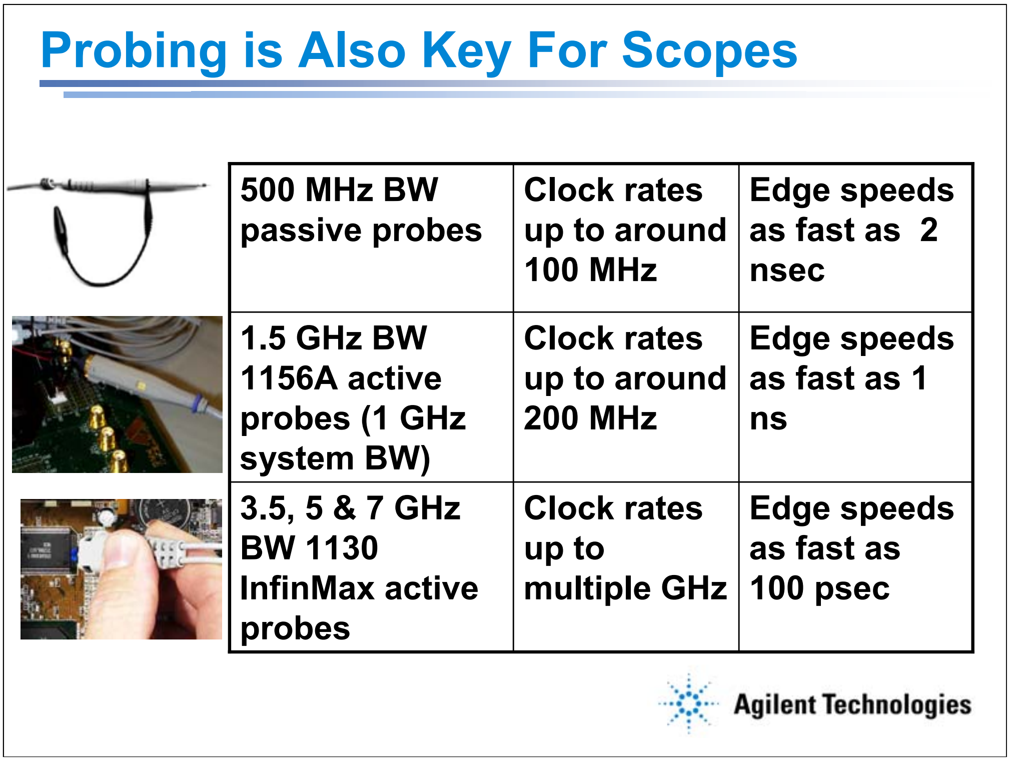
- No connector - cost is reduced
- Signal doesn’t go through a connector, loading is reduced. (~0.7pF)
- Signal routing is improved.
- Signal Density is increased

Agilent Technologies

One important, new category of logic analyzer probes is called “Connector-Less” probing. This is the newest technology in logic analyzer probing and presents some of the most attractive options to the end user.

The concept of a connector- less probe is that the user only puts down landing pads on his PCB. Traces are routed to or through the pads. A retention module is used to align the probe tip with the pads and provide mechanical stability. The logic analyzer probe is then attached to the board with the assistance of the retention module and makes contact with the pads on the board.

The advantages of connector- less probing are that since the signals do not travel through a connector to get to the logic analyzer probe tip, there is less capacitive loading on the target. For example, the E5378A probe discussed earlier had an equivalent lumped capacitance of 1.5pF. Connectorless probes are on the order of 0.7pF, a 2x improvement. Also attractive about this new technology is that the user does not need to load connectors onto the target PCB, reducing the cost of the final assembly. Finally, improvements have been made to the rout e-ability of connector- less probes. The user can now easily route directly through the probe pads without making layer changes. Since the capacitance is so low and there is no connector, the logic analyzer footprint can be left in the design on the final release of the PCB without any electrical degradation to system performance.



Probing is Also Key For Scopes

500 MHz BW passive probes	Clock rates up to around 100 MHz	Edge speeds as fast as 2 nsec
1.5 GHz BW 1156A active probes (1 GHz system BW)	Clock rates up to around 200 MHz	Edge speeds as fast as 1 ns
3.5, 5 & 7 GHz BW 1130 InfinMax active probes	Clock rates up to multiple GHz	Edge speeds as fast as 100 psec

Agilent Technologies

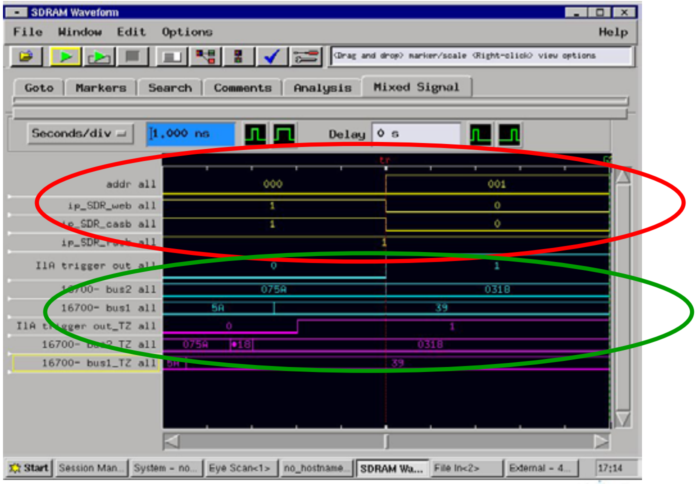
An important part of debugging FPGA-based systems is the ability to make oscilloscope measurements both at the FPGA boundary and at other points in the system. Passive probes have their place, but at the faster clock rates and edge speeds scope bandwidths of 1 GHz or better become necessary. For the 1 GHz Infiniium scopes, the 1156A active probes are the right choice to maintain a full 1 GHz system bandwidth for single-ended measurements. A wide range of connection accessories with documented performance give lots of options for how to attach to various points in the system. Soon there will be new differential active probes to maintain full system bandwidth as well.

For the very fast signals, the 54850 series of Infiniium scopes are a best fit. For these scopes, the InfiniMax single-ended and differential probes and various high bandwidth connection accessories offer unmatched performance.

Calculations for limits in edge speeds are based on system bandwidths necessary for around a 5% rise time measurement error. The 500 MHz and 1 GHz scope models assumed a gaussian scope frequency response, and the very high bandwidth scope models (> 1 GHz bandwidth) assumed a flat scope frequency response. Designers may be comfortable with higher errors than 5%, especially for timing measurements, which would extend the use range.

A “System” View

ILA Core signals imported and correlated to logic analyzer capture



ILA Core Signals

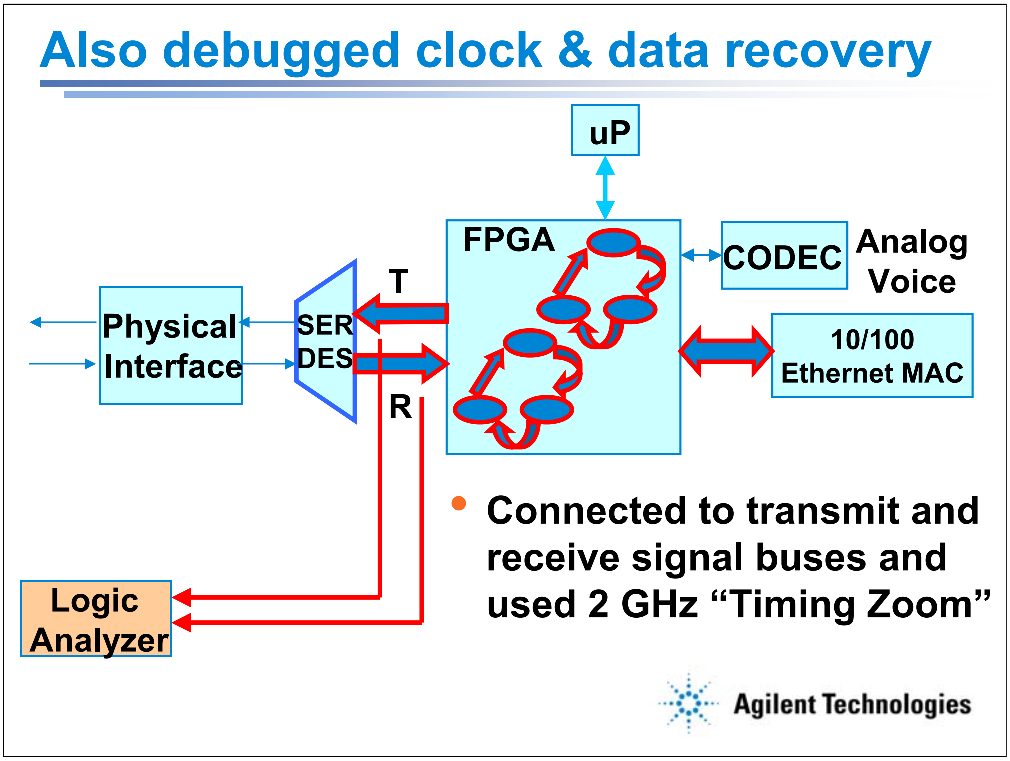
External LA Signals

Agilent 16702B display

Agilent Technologies

When making measurements in the system, it can be helpful to see both the internal FPGA signals and other system signals. Earlier we saw a view of SDRAM signals internal to a Xilinx Virtex II Pro FPGA. What about trying to view other system signals in context with those internal FPGA signals?

Xilinx and Agilent have worked together to make this easy. By dedicating one pin for a common debug signal, Xilinx ChipScope Pro can export internal FPGA signals by LAN to Agilent system logic analyzers. Here's a logic analyzer view of ChipScope data time correlated the logic analyzer captured data.



Also debugged clock & data recovery

- Connected to transmit and receive signal buses and used 2 GHz “Timing Zoom”

Agilent Technologies

What about an example of looking at the boundary of an FPGA?

If we look back to that earlier fiber to the home ATM network example, there was another interesting debug situation on this same design, that involved the debug of the clock and data recovery circuits on the front end. These were measurements taken outside the FPGA on its boundary, where the parallel transmit and receive buses interfaced with the FPGA.

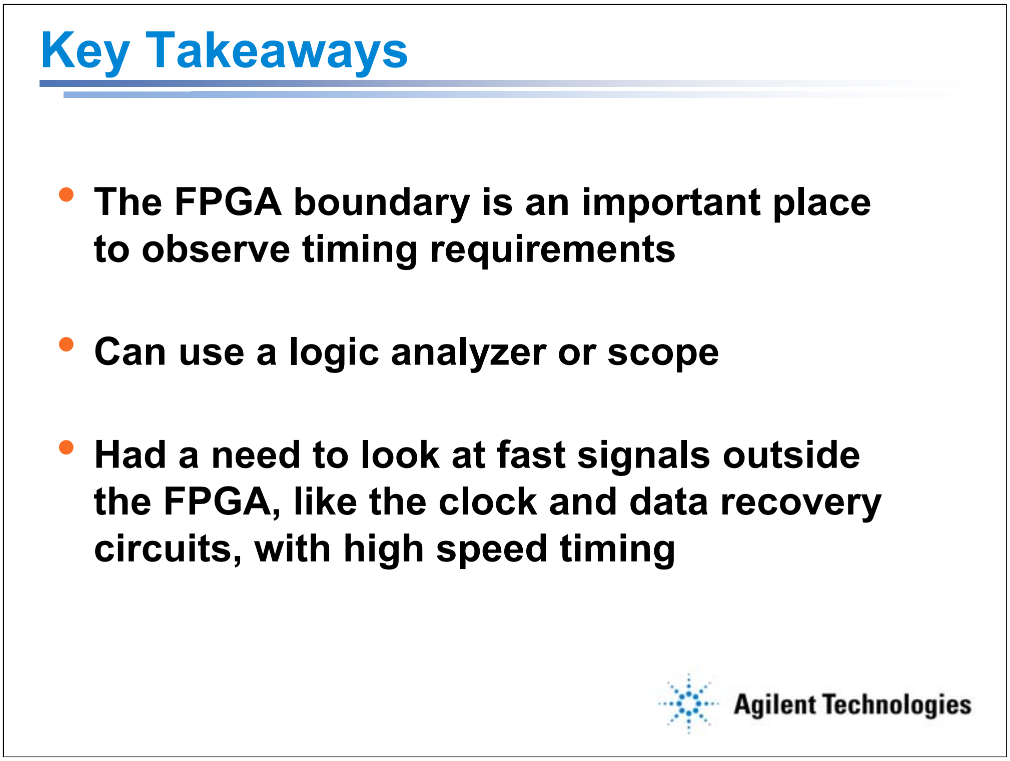
They were having a failure, and trying to see the conditions surrounding this error. They put the transmitter on one logic analyzer machine and the receiver on the other, and gathered a deep trace to begin looking for the nature of the problem. They ended up using a single logic analyzer machine and its 2 GHz Timing Zoom capture to observe clock phase with respect to data and the conditions where errors were occurring.

Debug Approach

- Used 2 GHz Timing Zoom with one machine to see clock phase differences

- Watching timing of data with respect to the clocks that caused failure

Agilent Technologies



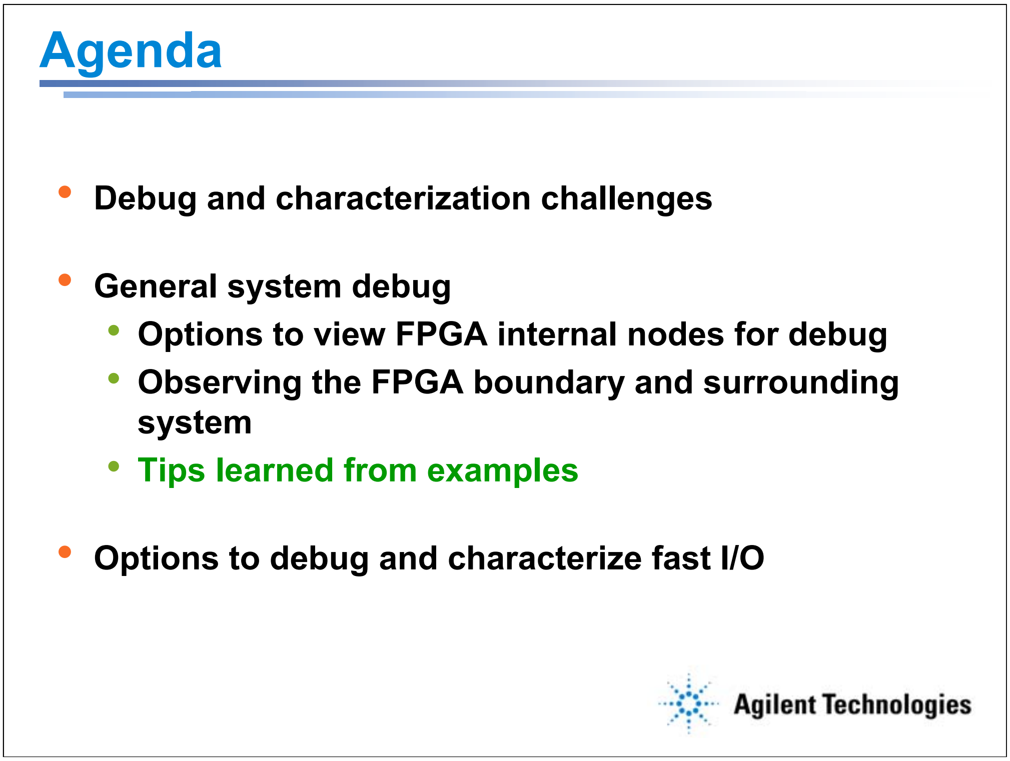
Key Takeaways

- The FPGA boundary is an important place to observe timing requirements
- Can use a logic analyzer or scope
- Had a need to look at fast signals outside the FPGA, like the clock and data recovery circuits, with high speed timing

Agilent Technologies

A few takeaways from this example:

The designer had a need to look at signals outside the FPGA that needed to meet FPGA input specifications, likes the clock and data recovery circuit signals, with high-speed 2 gigahertz timing.



Agenda

- Debug and characterization challenges
- General system debug
  - Options to view FPGA internal nodes for debug
  - Observing the FPGA boundary and surrounding system
  - Tips learned from examples
- Options to debug and characterize fast I/O

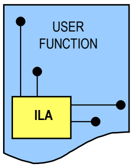
Agilent Technologies

Agenda

So we’ve looked at several customer debug examples and considered some of the options surrounding basic debug. What can we learn from this?

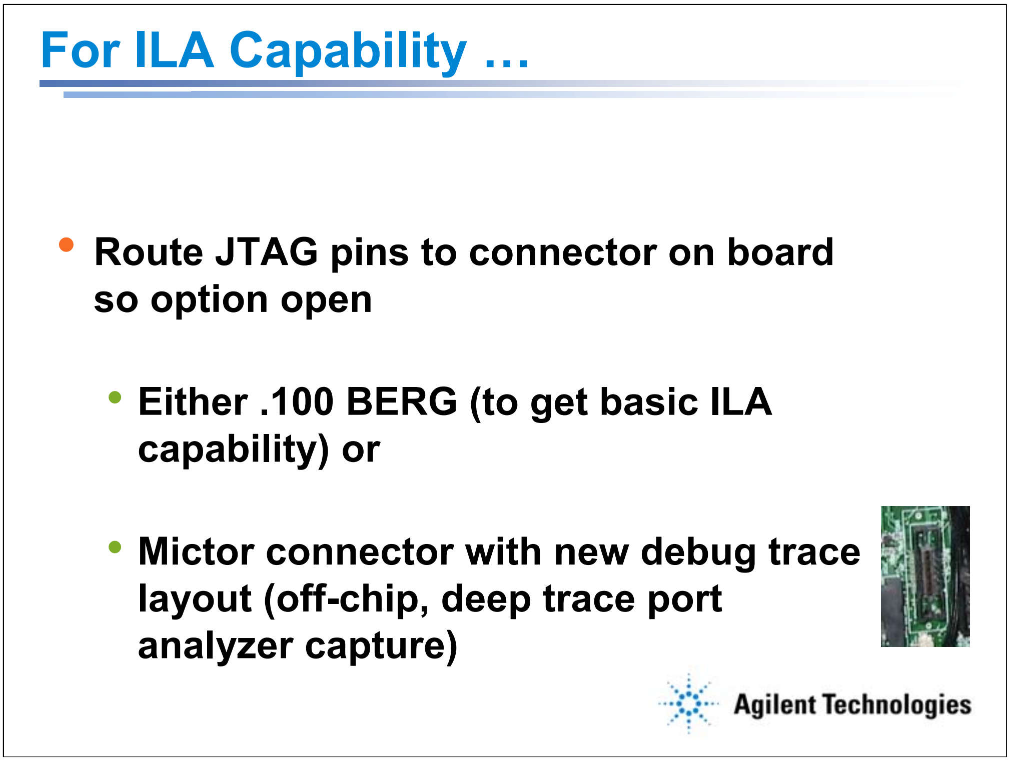
Tips Learned

- ILA cores and external logic analyzers are complimentary solutions



  - ILAs give lots of visibility (wide)
  - External LA gives deep capture, timing analysis
  - Still will want to dedicate some pins for LA

First, we’ve seen how embedded logic analyzer cores can be very helpful for visibility into FPGAs. We’ve also seen that external logic analyzers or scopes can be helpful too, given deep memory and the ability to do careful timing analysis. In fact, both approaches have their place and can be used together. Given that, it is still wise to dedicate some pins for this external trace.

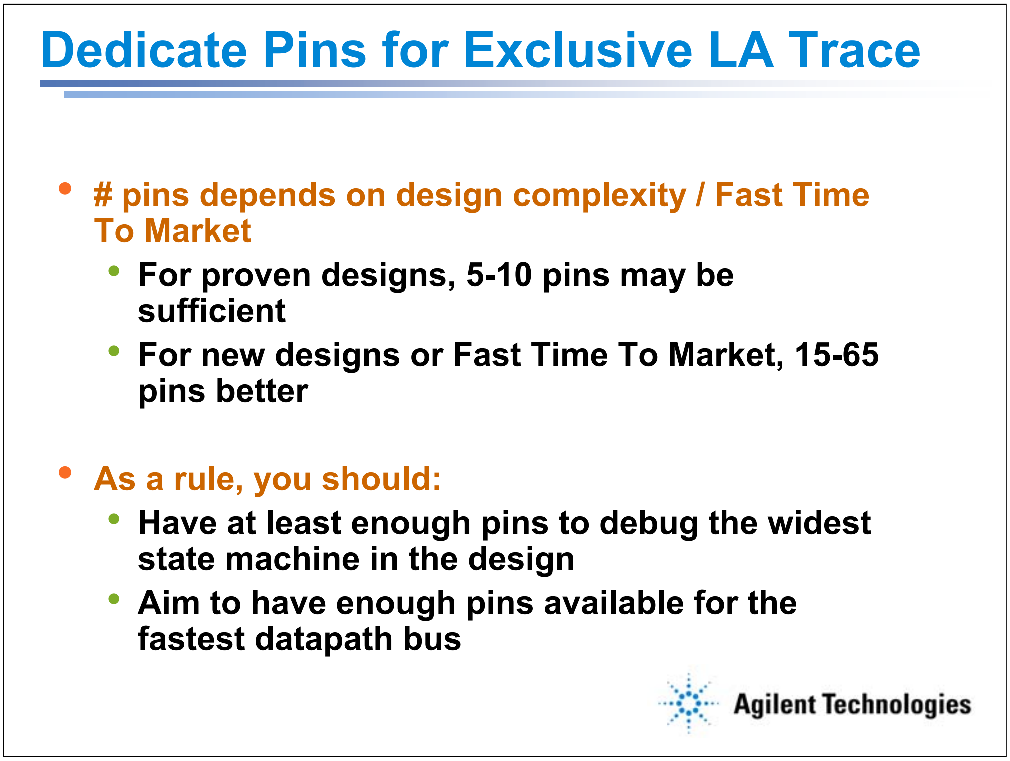


For ILA Capability …

- Route JTAG pins to connector on board so option open
  - Either .100 BERG (to get basic ILA capability) or
  - Mictor connector with new debug trace layout (off-chip, deep trace port analyzer capture)

Agilent Technologies

So for starters, plan so that you have options for ILA capability, with either a BERG connector for basic capture in Block RAM, or with a Mictor connector for the new off-chip approach with a Trace Port Analyzer.

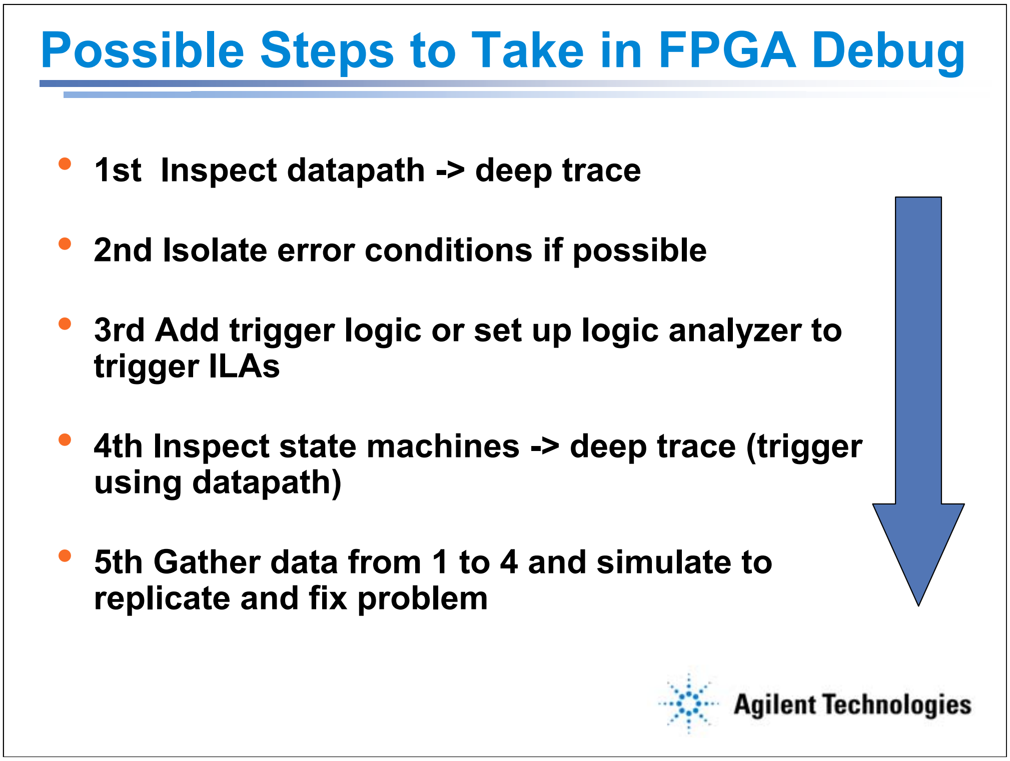


Dedicate Pins for Exclusive LA Trace

- # pins depends on design complexity / Fast Time To Market
  - For proven designs, 5-10 pins may be sufficient
  - For new designs or Fast Time To Market, 15-65 pins better
- As a rule, you should:
  - Have at least enough pins to debug the widest state machine in the design
  - Aim to have enough pins available for the fastest datapath bus

Agilent Technologies

Here are some possible ideas about the number of debug pins you might need. It was clear, from the examples we’ve discussed, that it was often critical to be able to have enough pins to see internal state machines, and to see datapaths. Then take advantage of the ways additional visibility is possible with the ILA cores.

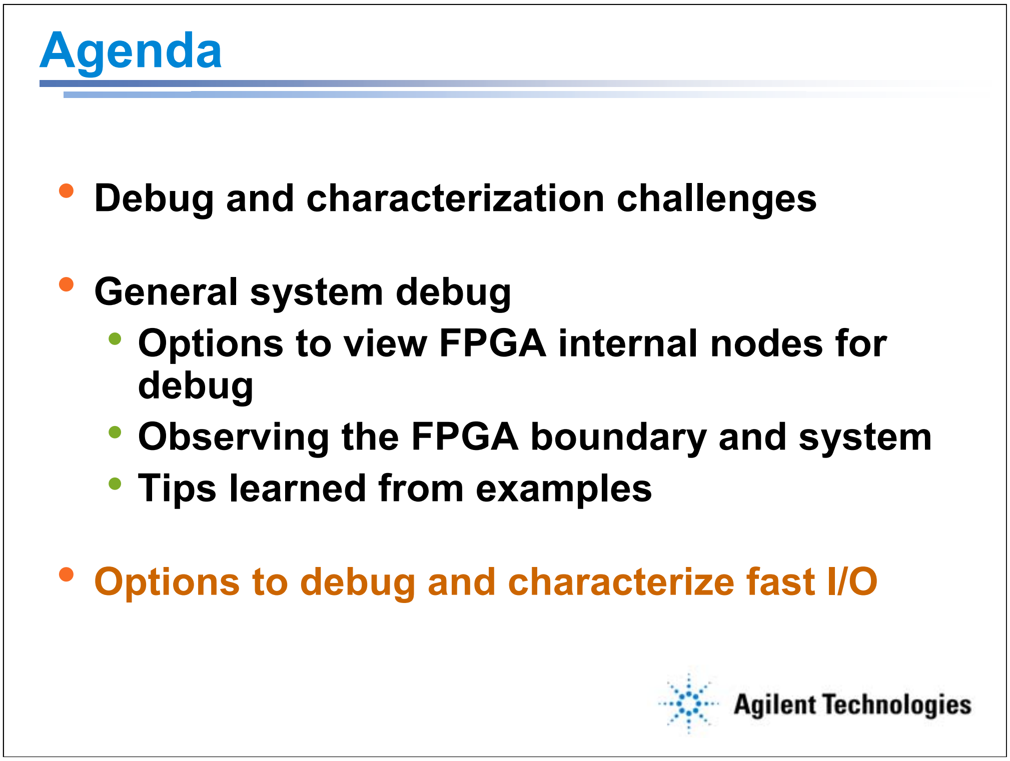


Possible Steps to Take in FPGA Debug

- 1st Inspect datapath -> deep trace
- 2nd Isolate error conditions if possible
- 3rd Add trigger logic or set up logic analyzer to trigger ILAs
- 4th Inspect state machines -> deep trace (trigger using datapath)
- 5th Gather data from 1 to 4 and simulate to replicate and fix problem

Agilent Technologies

Looking at a variety of debug situations, a common theme seemed to form in the way people discovered fixes to their problems. Many times, designers saw a symptom of a problem on a bus, figured out how to isolate that symptom, and then defined a trigger so that they could look internally to the FPGA to see state machine activity. With that kind of visibility, they could form a hypothesis to the problem and simulate it. Then the fix would become obvious.



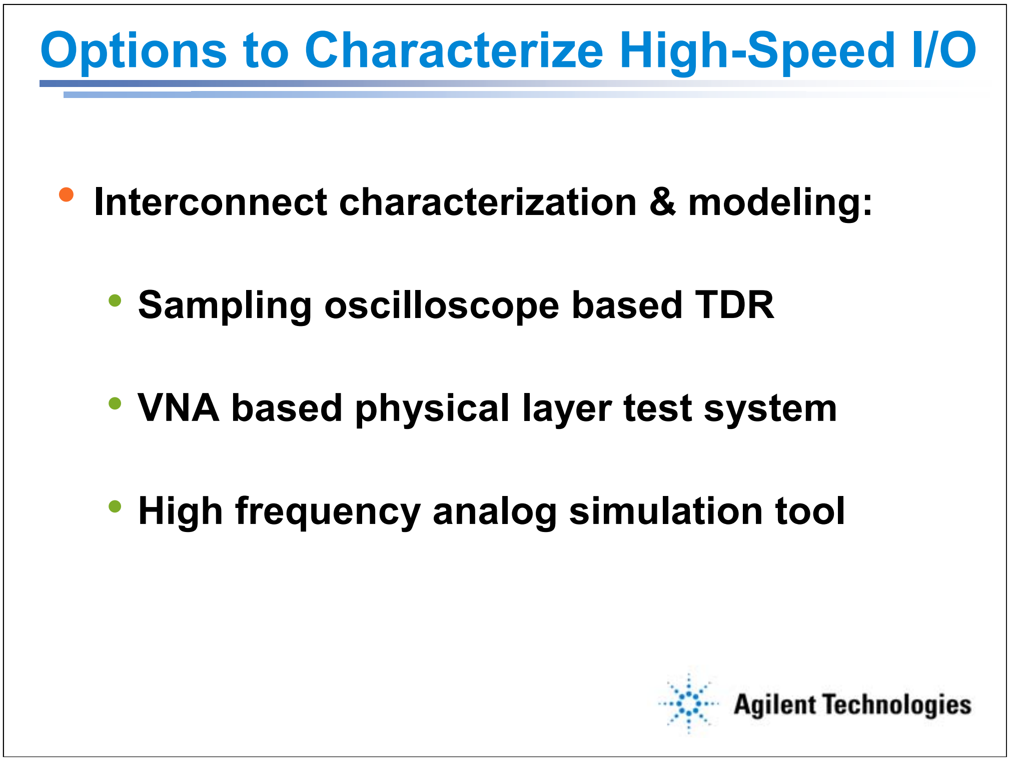
Agenda

- Debug and characterization challenges
- General system debug
  - Options to view FPGA internal nodes for debug
  - Observing the FPGA boundary and system
  - Tips learned from examples
- Options to debug and characterize fast I/O

Agilent Technologies

Agenda

So what about approaches when dealing with high speed I/O debug and characterization?

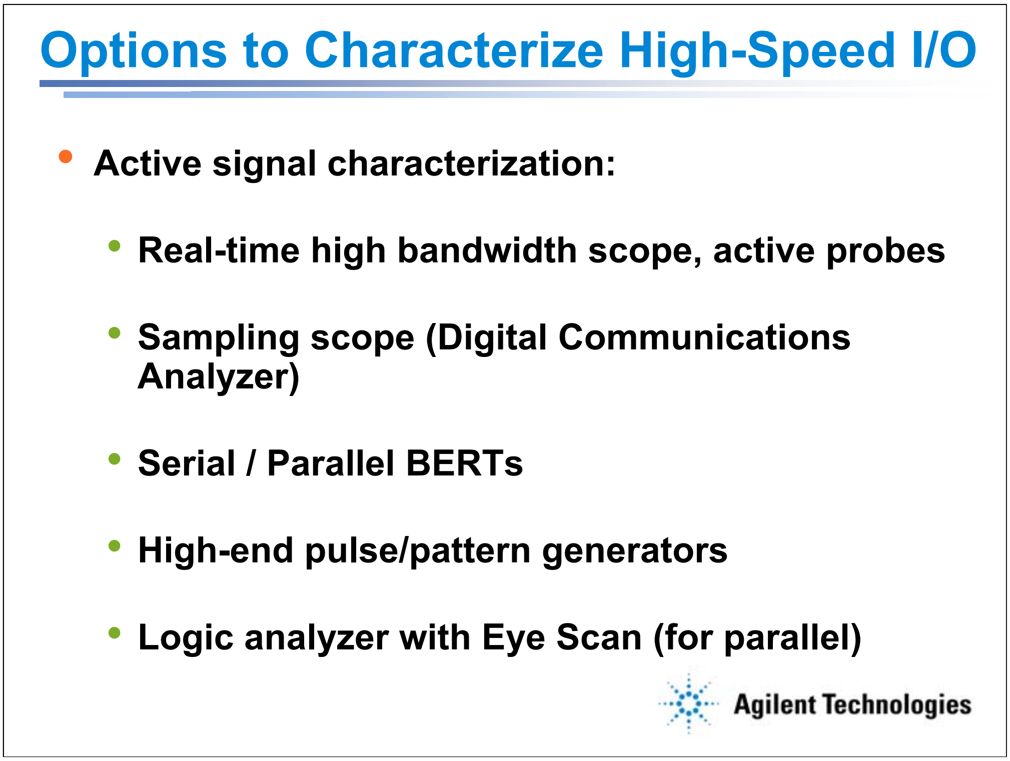


Options to Characterize High-Speed I/O

- Interconnect characterization & modeling:
  - Sampling oscilloscope based TDR
  - VNA based physical layer test system
  - High frequency analog simulation tool

Agilent Technologies

First, there are a number of options for looking at passive interconnect to understand transmission line characteristics. These vary from sampling oscilloscope based solutions like Time Domain Reflectometry, or TDR, to Vector Network Analyzer based solutions. There are also simulation tools very useful to understand high frequency analog effects to fast digital signals.

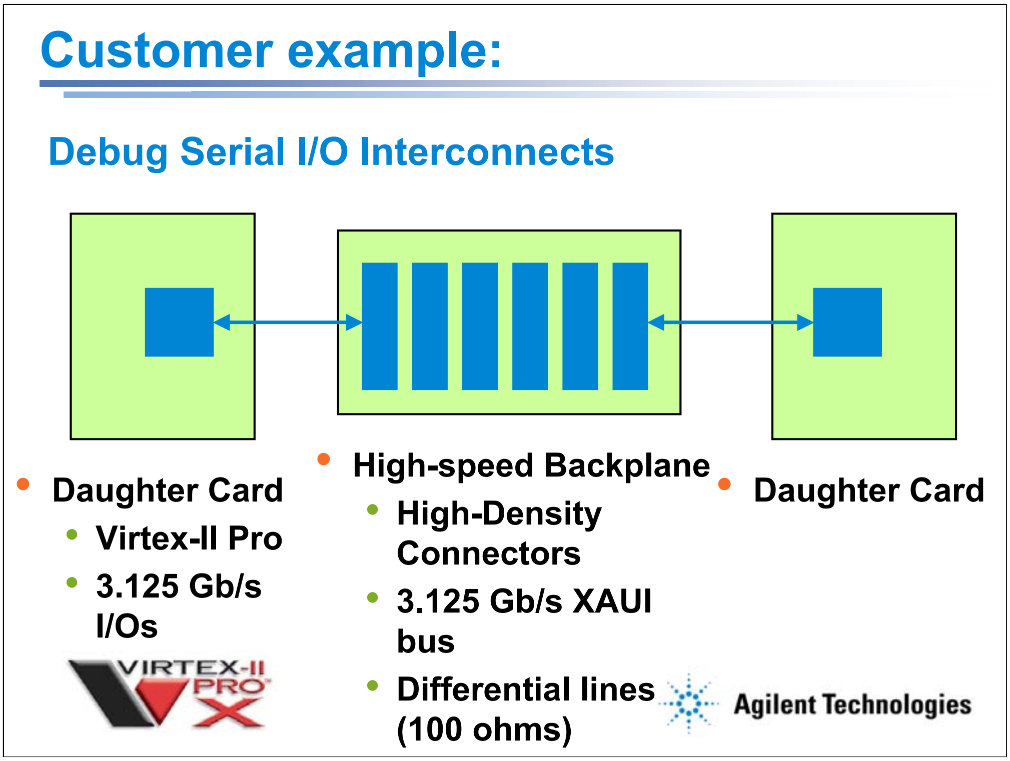


Options to Characterize High-Speed I/O

- Active signal characterization:
  - Real-time high bandwidth scope, active probes
  - Sampling scope (Digital Communications Analyzer)
  - Serial / Parallel BERTs
  - High-end pulse/pattern generators
  - Logic analyzer with Eye Scan (for parallel)

Agilent Technologies

To understand whether a high-speed I/O is working properly, you certainly can’t only look at the passive interconnect. That’s where a wide variety of measurements come into play to look at fast I/O signals and the transfer of data in a system.



Customer example:

Debug Serial I/O Interconnects

- Daughter Card
  - Virtex-II Pro
  - 3.125 Gb/s I/Os
- High-speed Backplane
  - High-Density Connectors
  - 3.125 Gb/s XAUI bus
  - Differential lines (100 ohms)
- Daughter Card

VIRTEX-II PRO X

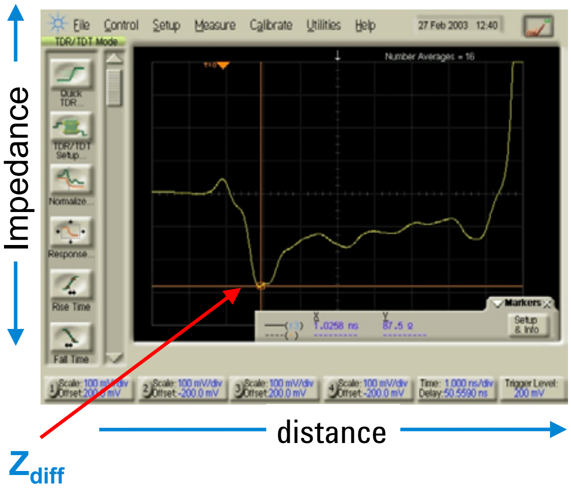
Agilent Technologies

Our final example crosses over from general debug over into high speed I/O debug and characterization. Here a company was designing the backplane structure for line cards in a router. It was going to run at 3.125 Gb/s XAUI.

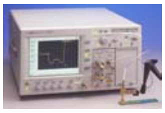
It was necessary to characterize the performance of the transmission line paths from the FPGA I/O output to the FPGA input on a second daughter card. A spec had to be met for impedance and crosstalk.

Insight With Time Domain Reflectometer

- Differential Impedance (Zdiff)
  - Measured 87 Ω at interface to backplane
  - Too low for spec (100 Ω)

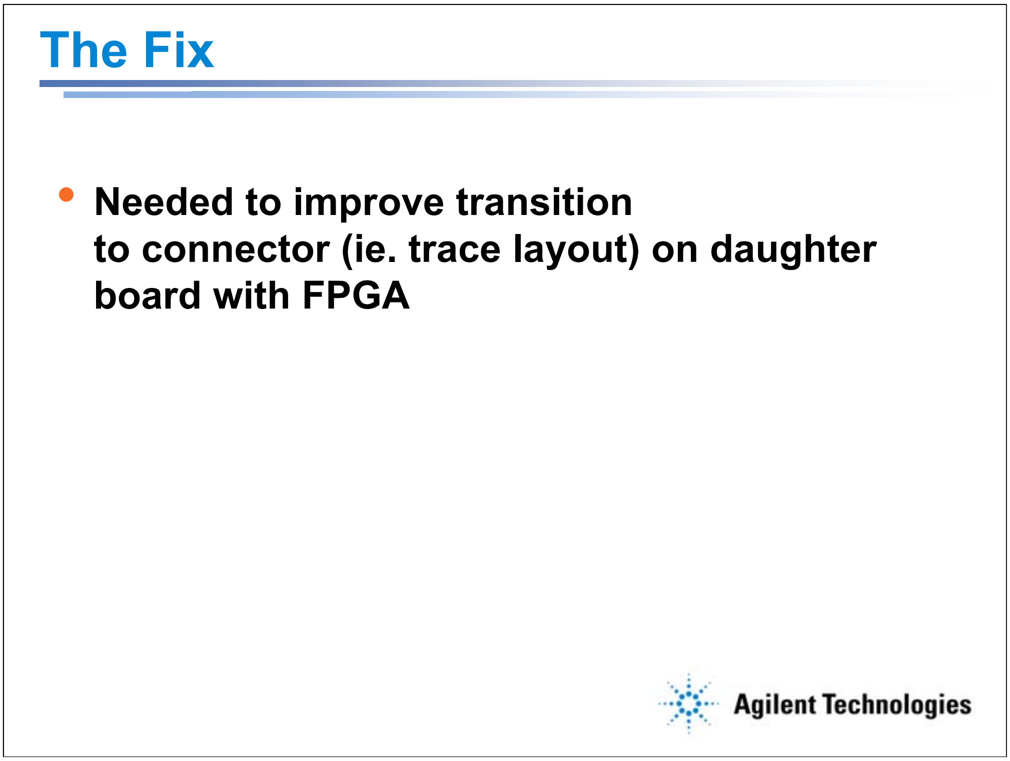


Impedance
File Control Setup Measure Calibrate Utilities Help
27 Feb 2003 12:40
TDR/TDT Mode
Number Averages = 16
Quick TDR
TDR/TDT Setup
Normalize
Response
Rise Time
Fall Time
Markers
1.0258 ns
87.5 Ω
Setup & Info
Scale: 100 mV/div Offset: 200.0 mV
Scale: 100 mV/div Offset: -200.0 mV
Scale: 100 mV/div Offset: 200.0 mV
Scale: 100 mV/div Offset: -200.0 mV
Time: 1.000 ns/div Delay: 50.5590 ns
Trigger Level: 200 mV
distance
Z_{diff}



Agilent Technologies

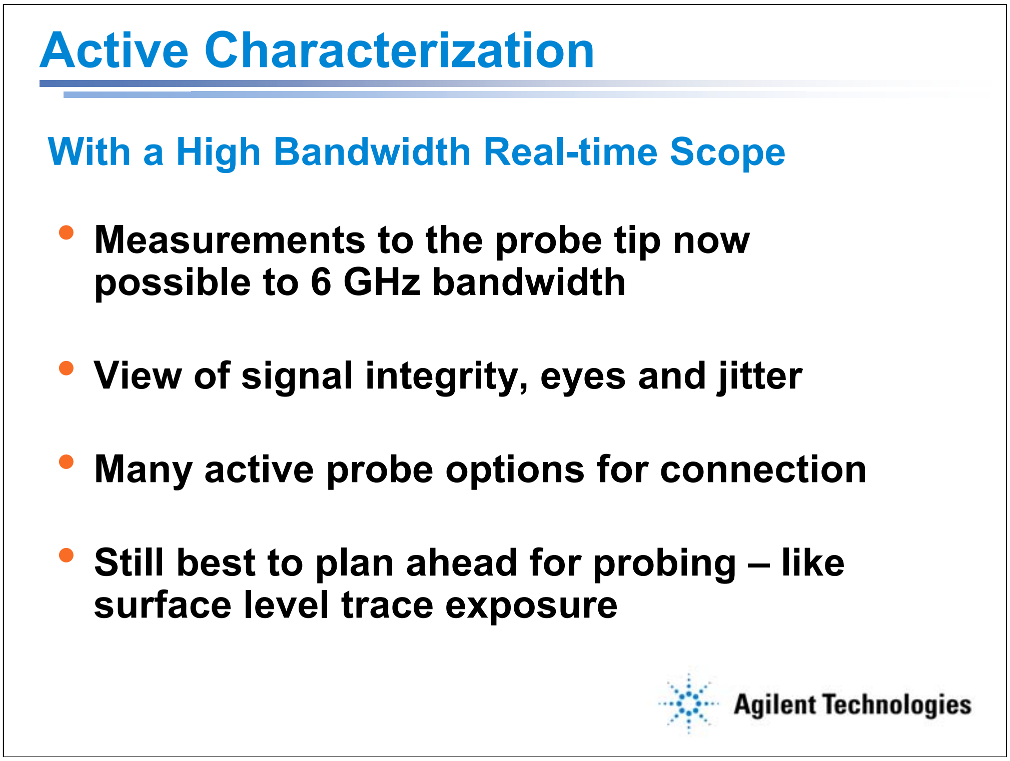
The TDR showed where the impedance was too low, and pointed to the location where capacitance needed to be designed out of a transition from the daughter board PC board trace out to a connector.



The Fix

- Needed to improve transition to connector (ie. trace layout) on daughter board with FPGA

The measurements pointed to the need for a better termination at the far end.



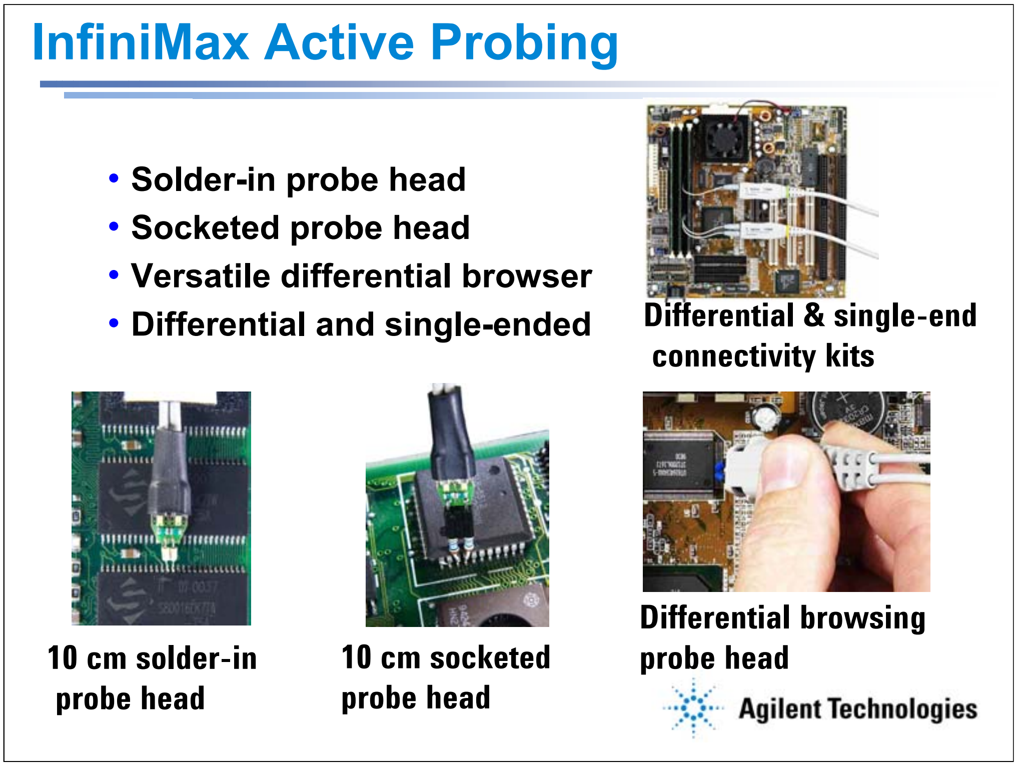
Active Characterization

With a High Bandwidth Real-time Scope

- Measurements to the probe tip now possible to 6 GHz bandwidth
- View of signal integrity, eyes and jitter
- Many active probe options for connection
- Still best to plan ahead for probing – like surface level trace exposure

Agilent Technologies

New measurements are now available with the advent of high bandwidth real-time scopes and new technology differential active probes. Some of the planning for this can include routing some key signals on the surface so if it is necessary, the LPI layer can be scratched back to expose a differential pair, or the LPI layer can be omitted from an area intended for probing.



InfiniMax Active Probing

- Solder-in probe head
- Socketed probe head
- Versatile differential browser
- Differential and single-ended

Differential & single-end connectivity kits

10 cm solder-in probe head

10 cm socketed probe head

Differential browsing probe head

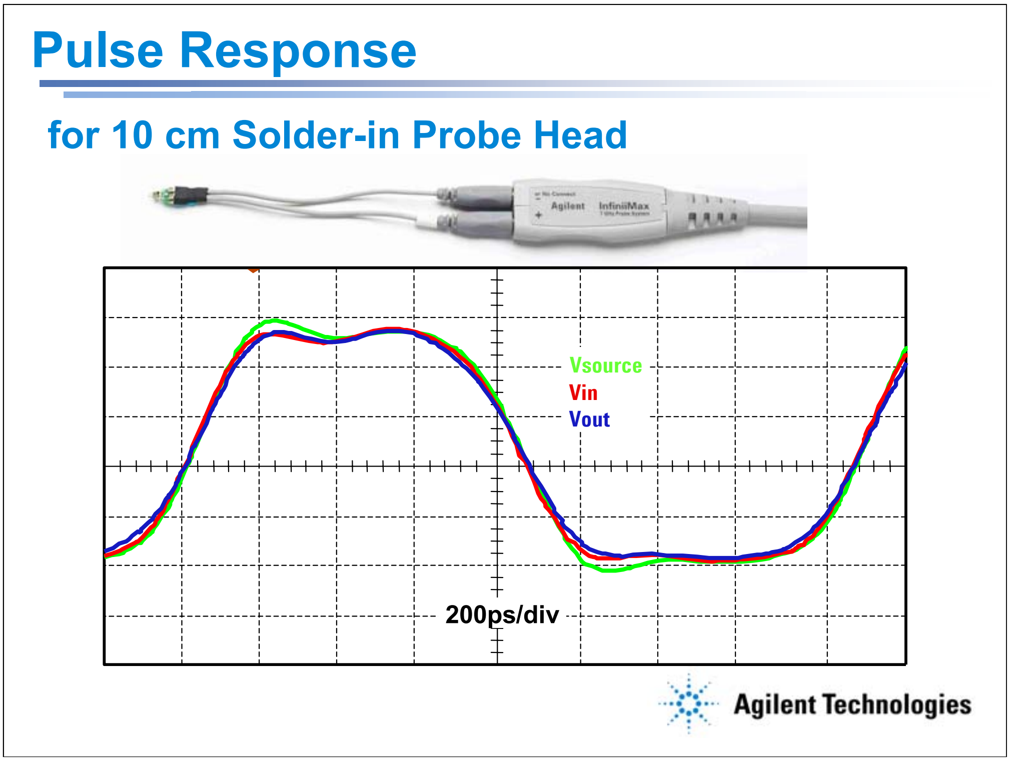
Agilent Technologies

Higher Bandwidth Connection Solutions

Use models for the InfiniiMax active probe system are similar to traditional active probes, including “browsing” connections, 10cm solder-in connections, and 10cm socketed connections. In addition, the InfiniiMax probe system includes both single-ended and differential measurement use models. The big difference between the InfiniiMax probe system and traditional active probe systems is probe system bandwidth. Even with long connection use models, there is no bandwidth loss penalty.

For high-speed I/O on ball-grid array type FPGAs, it is probably not best to place vias under the FPGA. It turns out, if you do, you don’t gain too much for probing since there is still as much as a 14 mm trace from the package to the substrate. You’ll probably want to use a double termination (source and receiver), and route key signals to the surface so you can do mid-bus probing. One approach is to simply design a small area where those traces are not coated with LPI (insulated coating on top of board) for easy differential probe access. Alternatively, one can scratch off the LPI in area desired for probing.

Sometimes it’s best to probe back at a transmitter location, even though the receiver is inside the FPGA at the substrate point.



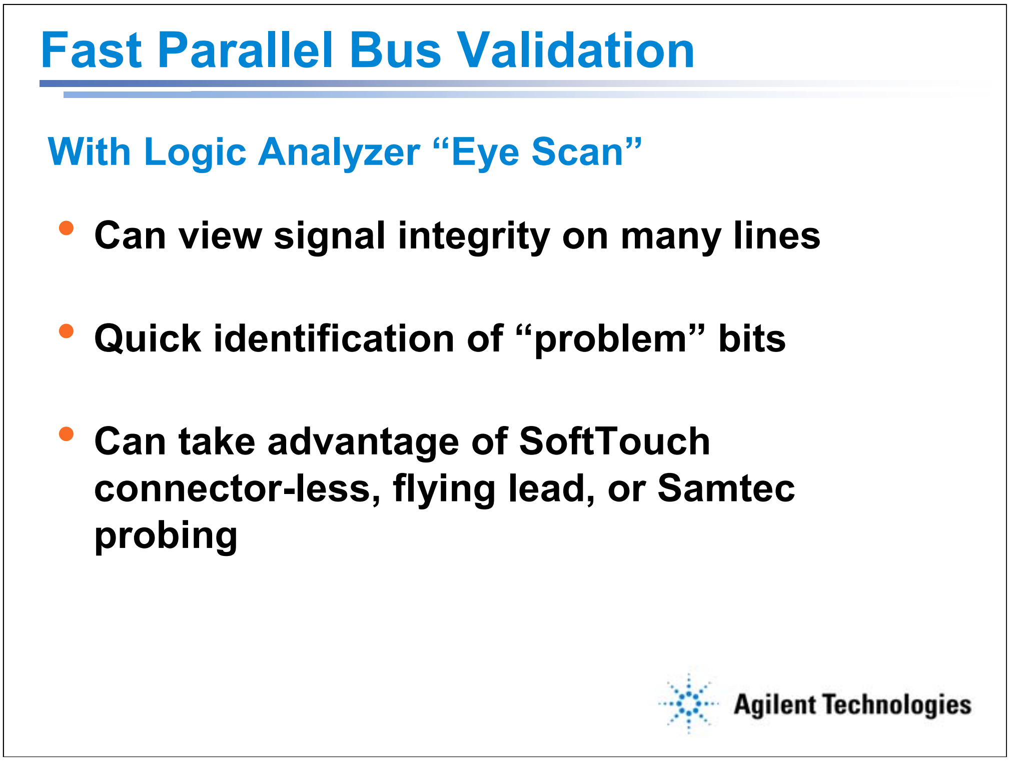
Pulse Response

for 10 cm Solder-in Probe Head

Agilent Technologies

Pulse Response for 10cm solder-in probe heads – 1.2GHz Clock, 100ps Risetime

These are the same waveforms shown in the previous slide for the 10cm solder-in probe head with all of the waveforms overlaid on top of one another. This is as good as it gets! No other scope and probe combination can duplicate these results. From any vendor.



Fast Parallel Bus Validation

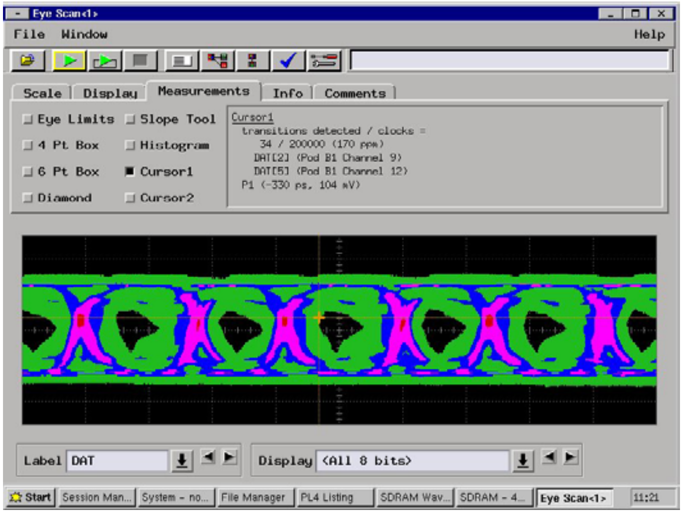
With Logic Analyzer "Eye Scan"

- Can view signal integrity on many lines
- Quick identification of "problem" bits
- Can take advantage of SoftTouch connector-less, flying lead, or Samtec probing

Agilent Technologies

Designers also face signal integrity validation on fast parallel buses, such as 622 MHz SPI 4.2 signals. Now there is a new approach to identifying problems. It's with a logic analyzer and called Eye Scan.

SPI 4.2 Bus Seen With Eye Scan



□ Eye Limits □ Slope Tool
□ 4 Pt Box □ Histogram
□ 6 Pt Box ■ Cursor1
□ Diamond □ Cursor2

```
Cursor1
 transitions detected / clocks =
    34 / 200000 (170 ppm)
   DAT[2] (Pod B1 Channel 9)
   DAT[5] (Pod B1 Channel 12)
 P1 (-330 ps, 104 mV)
```

Label DAT

Display <All 8 bits>

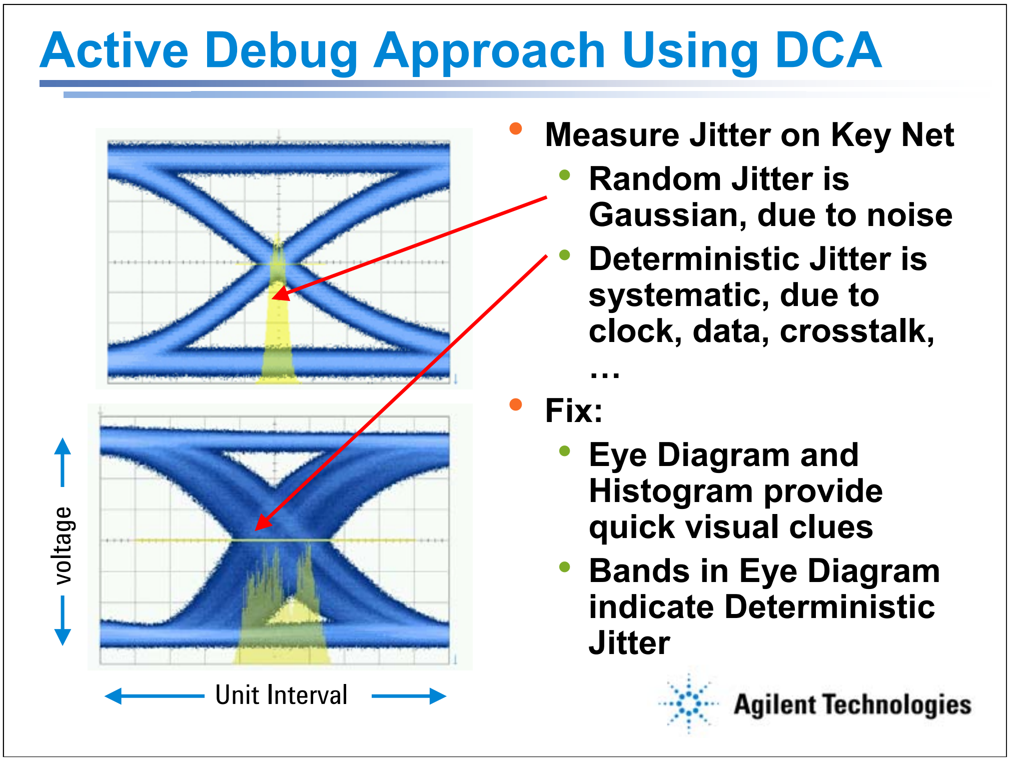
Agilent Technologies

Eye Scan is a third “mode” in a logic analyzer beyond “State” or “Timing” modes. With Eye Scan, the analyzer threshold is not just set to a “1 – 0” level, but rather is varied over a range of levels. For each level, the analyzer also adjust the time point of observation before and after trigger, and then counts the number of transitions it sees through the threshhold.

The result is that over many clocks, an eye diagram is built up with multiple GHz analog bandwidth, and looking at many signals.

Here, with a cursor, we could just touch the left had side of the eye opening, and see exactly “which” bits are closing off the eye.

This can point a designer immediately to problems on specific bits such as jitter, noise, crosstalk, or other signal integrity issues. Then a high bandwidth scope and active probe can be used to look more closely.

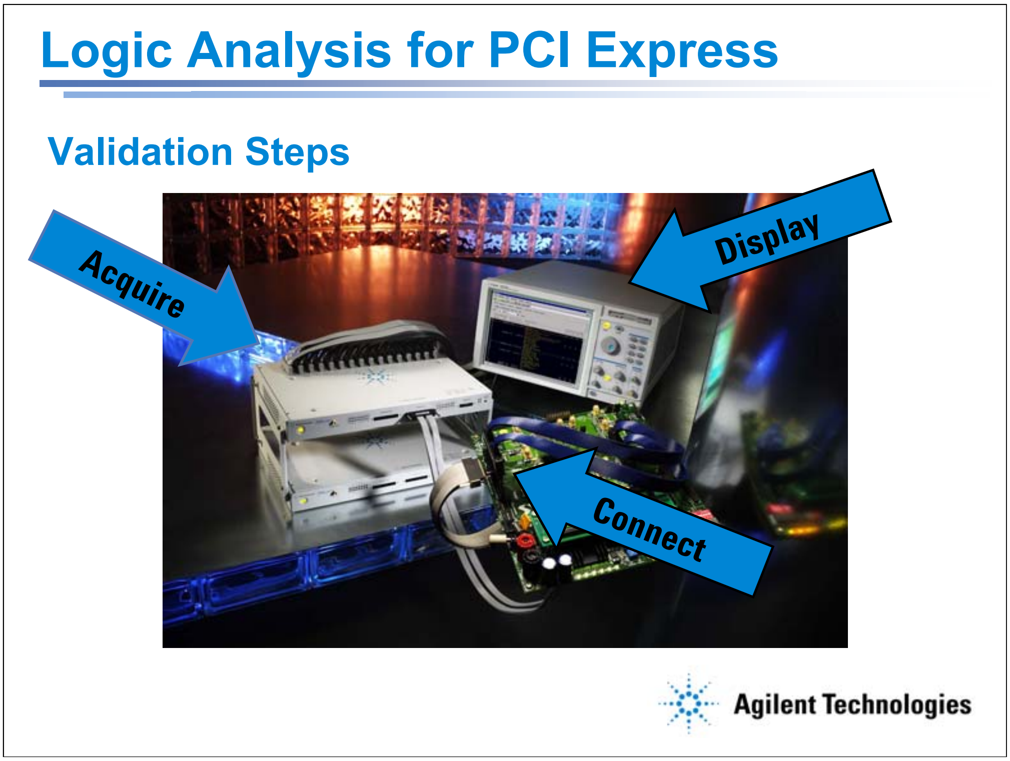


Active Debug Approach Using DCA

- Measure Jitter on Key Net
  - Random Jitter is Gaussian, due to noise
  - Deterministic Jitter is systematic, due to clock, data, crosstalk, …
- Fix:
  - Eye Diagram and Histogram provide quick visual clues
  - Bands in Eye Diagram indicate Deterministic Jitter

voltage
Unit Interval
Agilent Technologies

Earlier we used TDR and crosstalk measurements to look at a passive interconnect. It is also important to make some measurements on signals through the backplane and look at jitter. It is possible to separate out the random jitter and deterministic jitter with software tools. Here is an example where it is obvious that there is data dependent, deterministic jitter that could pose a problem.



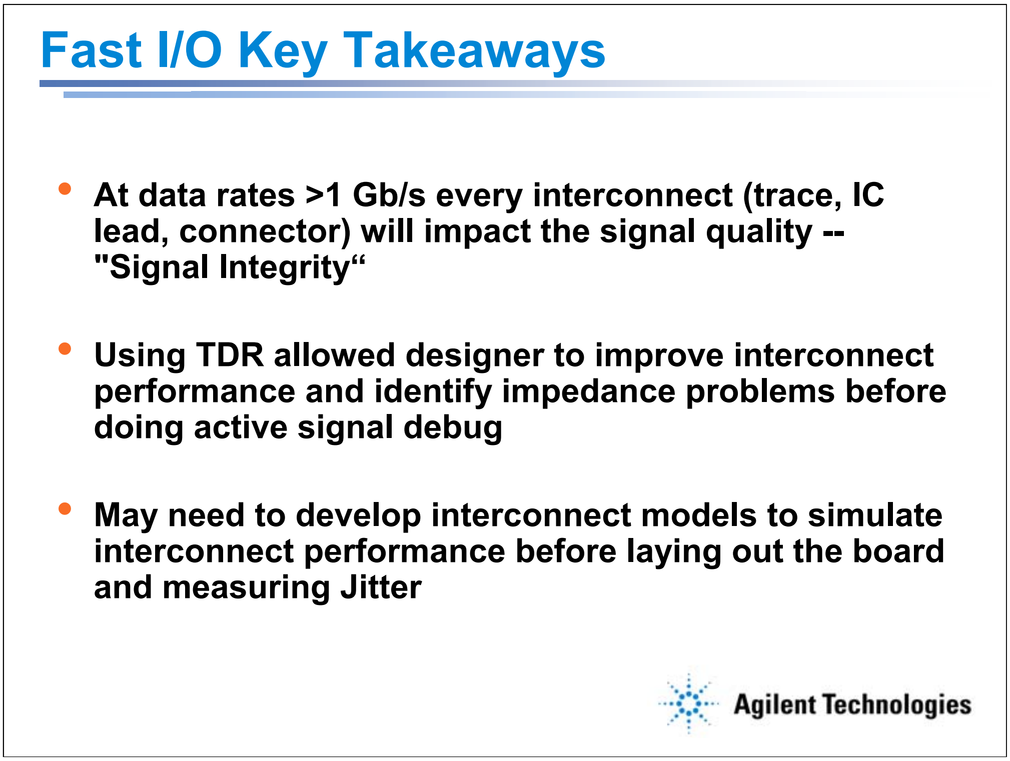
Logic Analysis for PCI Express

Validation Steps

Agilent Technologies

With some of the fastest serial I/Os, such as PCI Express, dedicated solutions are necessary to validate the bus or reveal problems. An example is the front end probe required to de-multiplex the fast PCI Express signal into slower, parallel signals that the logic analyzer can consume. Such a probe includes sophisticated triggering capabilities as well, thus reserving trigger resources inside the logic analyzer frame and acquisition module.

This solution uses the SoftTouch technology to do either mid-bus probing to a field of pads or an attachment into a PCI Express slot.



Fast I/O Key Takeaways

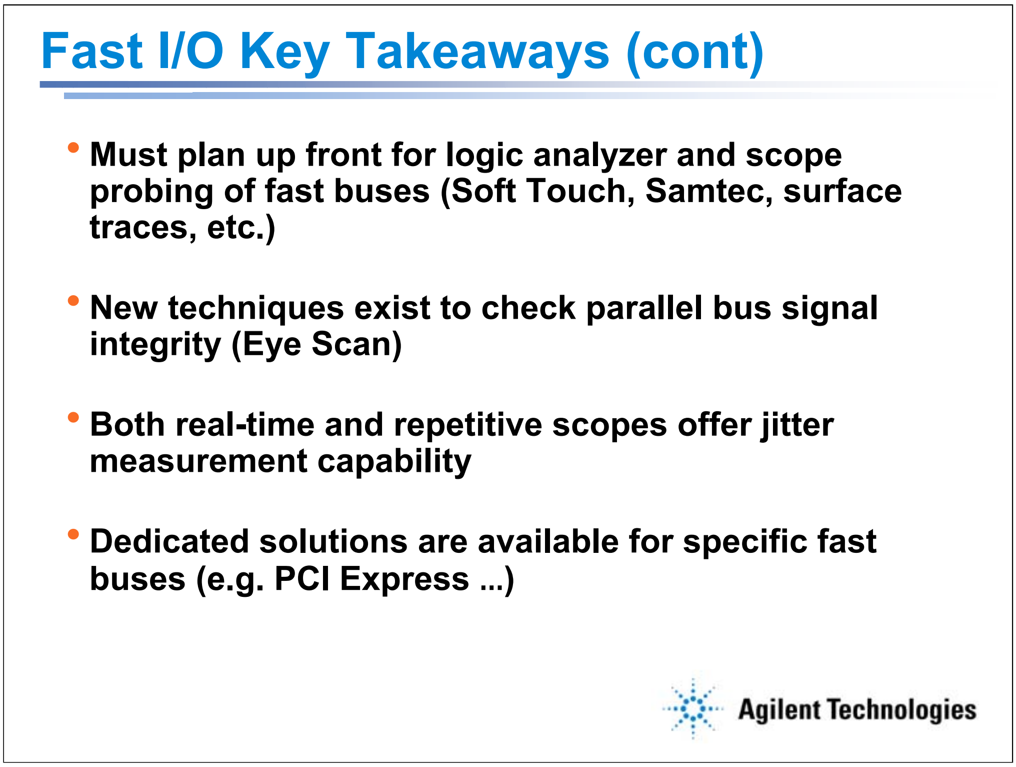
- At data rates >1 Gb/s every interconnect (trace, IC lead, connector) will impact the signal quality -- "Signal Integrity“

- Using TDR allowed designer to improve interconnect performance and identify impedance problems before doing active signal debug

- May need to develop interconnect models to simulate interconnect performance before laying out the board and measuring Jitter

Agilent Technologies

There were a number of important takeaways for I/O characterization..



Fast I/O Key Takeaways (cont)

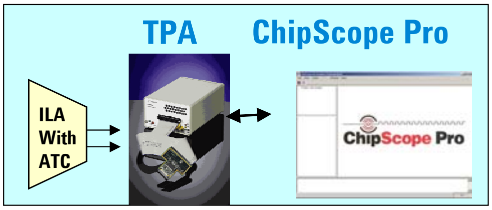
- Must plan up front for logic analyzer and scope probing of fast buses (Soft Touch, Samtec, surface traces, etc.)
- New techniques exist to check parallel bus signal integrity (Eye Scan)
- Both real-time and repetitive scopes offer jitter measurement capability
- Dedicated solutions are available for specific fast buses (e.g. PCI Express ...)

Takeaways continued.

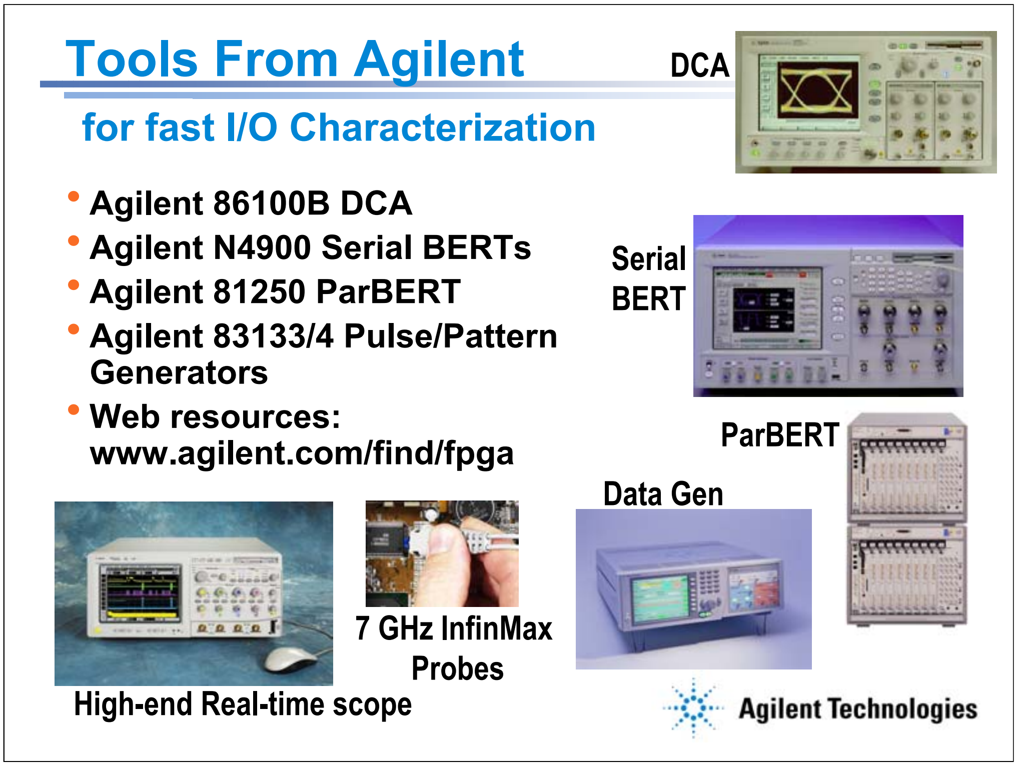
Tools From Agilent

For General FPGA Debug

- Agilent 16702B logic analysis system
- Agilent Infiniium Mixed Signal Oscilloscopes
- Agilent E5904B FPGA Trace Port Analyzer
- Application resources: www.agilent.com/find/fpga



When it comes to basic FPGA system debug, Agilent provides a variety of tools to help the designer of an FPGA based system, from the new Trace Port Analyzer to see internal nodes, to Mixed Signal Oscilloscopes for basic viewing of system activity, to logic analyzers with sophisticated triggering and data interpretation to point the designer to the root cause of their problems.



Tools From Agilent

for fast I/O Characterization

- Agilent 86100B DCA
- Agilent N4900 Serial BERTs
- Agilent 81250 ParBERT
- Agilent 83133/4 Pulse/Pattern Generators
- Web resources: www.agilent.com/find/fpga

DCA

Serial BERT

ParBERT

Data Gen

7 GHz InfinMax Probes

High-end Real-time scope

Agilent Technologies

When it comes to systems with fast I/Os, Agilent provides a variety of tools to help the designer of an FPGA based system, from the CSA to parallel bit error rate testers.