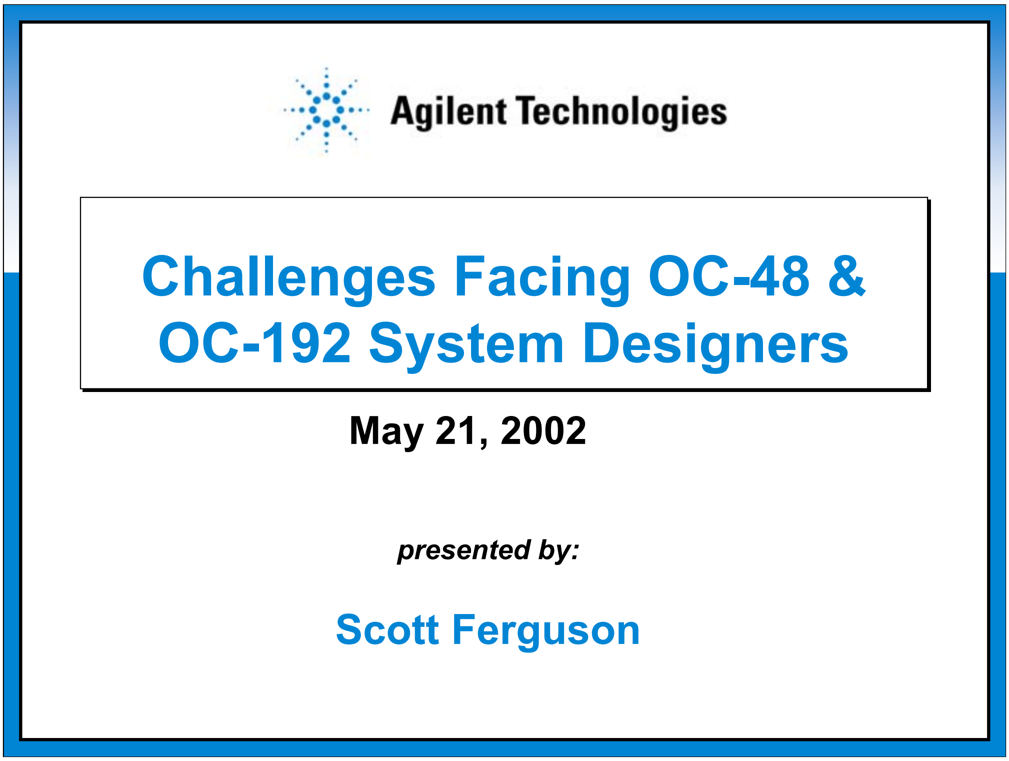


Agilent Technologies

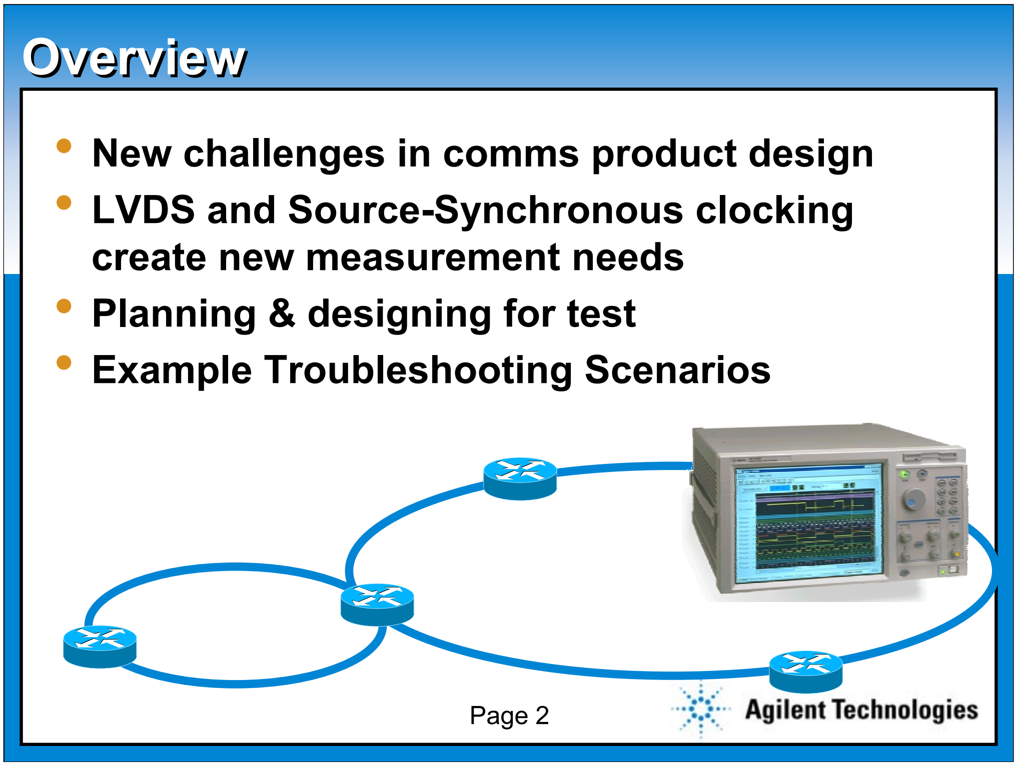
Challenges Facing OC-48 & OC-192 System Designers

May 21, 2002

presented by:

Scott Ferguson

Differential buses are increasingly popular in routers and switches because they offer higher speeds, lower cross talk, and less switching noise than single ended topologies. However, differential buses are significantly more complex than equivalent single ended designs when simulating, laying out, and measuring their performance. For example, choosing the optimum differential conductor placement and topology involves trading off a significant number of variables. Several real life challenges are identified and dealt with, presenting new options to designers. Solutions are based on using frequency- and time-domain-based measurements to characterize the bus and high-frequency simulation tools to model and improve the design. The methodologies employed are geared toward understanding how to properly design a differential bus for first pass success.



Overview

- New challenges in comms product design
- LVDS and Source-Synchronous clocking create new measurement needs
- Planning & designing for test
- Example Troubleshooting Scenarios

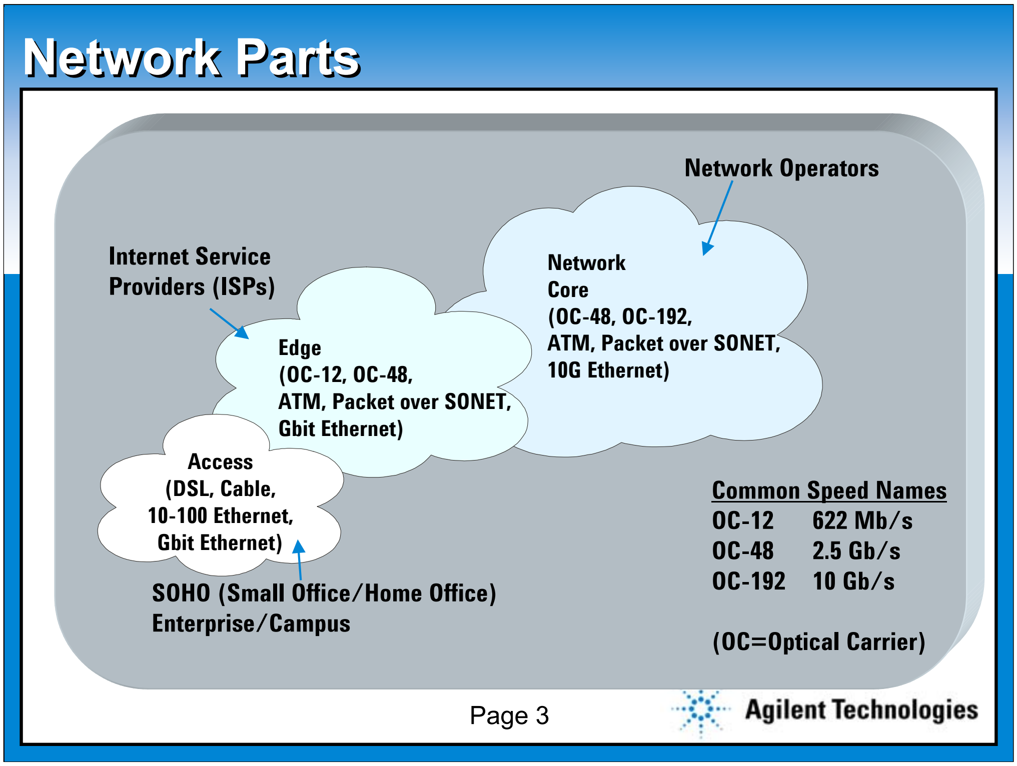
Agilent Technologies

Economic pressures in the networking and telecom industries have created new challenges for designers of network infrastructure equipment. It has been reported that telecom and network providers are only spending money on equipment that saves them money these days. Therefore constraints on floor space and power consumption become more important than in the past. Products need to be optimized for port density, heat dissipation, and power consumption per unit bandwidth.

New electronics technologies such as Low Voltage Differential Signaling (LVDS) and Source-Synchronous clocking allow a much greater amount of data throughput per unit power consumed and per IC pin. These technologies are therefore making inroads into commercial networking semiconductors and systems, all the way from the high-speed core network (where high speed is king) to the outer edge and access devices (where port density, power consumption and air conditioning capacity rule).

While this paradigm shift creates new opportunities for improvement, it also provides new challenges in product development. New test methodologies are required, and they must be considered at design time to prevent re-cuts of circuit boards and schedule slips.

This presentation will describe the new technologies, introduce test products that meet the new challenge, and describe a few measurement scenarios that put the new tools to work.



Network Parts

Network Operators

Internet Service Providers (ISPs)

Network Core
(OC-48, OC-192,
ATM, Packet over SONET,
10G Ethernet)

Edge
(OC-12, OC-48,
ATM, Packet over SONET,
Gbit Ethernet)

Access
(DSL, Cable,
10-100 Ethernet,
Gbit Ethernet)

SOHO (Small Office/Home Office)
Enterprise/Campus

Common Speed Names	
OC-12	622 Mb/s
OC-48	2.5 Gb/s
OC-192	10 Gb/s

(OC=Optical Carrier)

Agilent Technologies

The communication network can be divided into 3 main areas:

- The Core
- The Edge
- Access

Access devices provide broadband connections to the home, or aggregate low-speed traffic inside a company premises into a faster link to the outside world. At the Edge, medium-speed (622 Mbit/sec up to 2.5 Gbit/sec) traffic from businesses and ISP's is combined into high-speed links (up to 10 Gbits/second) and propagated through the core network.

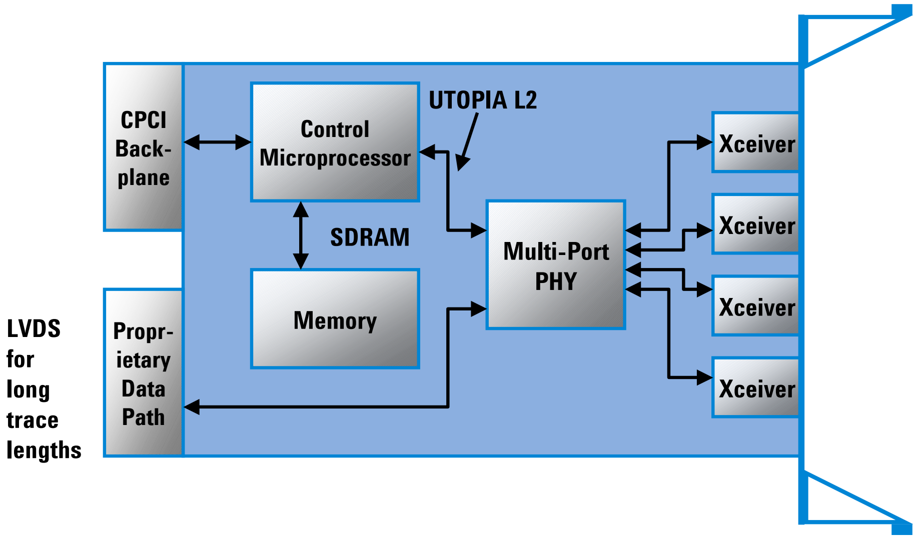
The products at the Access points (DSL Access Multiplexers or DSLAMS, Cable Modem Termination Systems, Enterprise Routers) have many different needs than the big-iron core routers, although the basic building blocks are very similar.

- Access switching centers pack as many ports as possible into a small building or even a box on the corner. With so many connections in a small space, keeping things cool can be tricky, as well as keeping down the cost of powering and servicing all those connections. However speeds are measured in hundreds of kilobits per second, so often standard microprocessors can be used to perform many functions.

- In the core network, the number of links is much smaller, but each link may run up to 10 Gbits/second. To feed such a fast link, a larger circuit is required, and therefore a Core router with less than 10 ports can take up as much space and power as a DSLAM with many ports.

These differences aside, the products all move packets from an input port to an appropriate output port, sometimes deciding to send more important ones before others, or deciding to reject traffic from hackers.

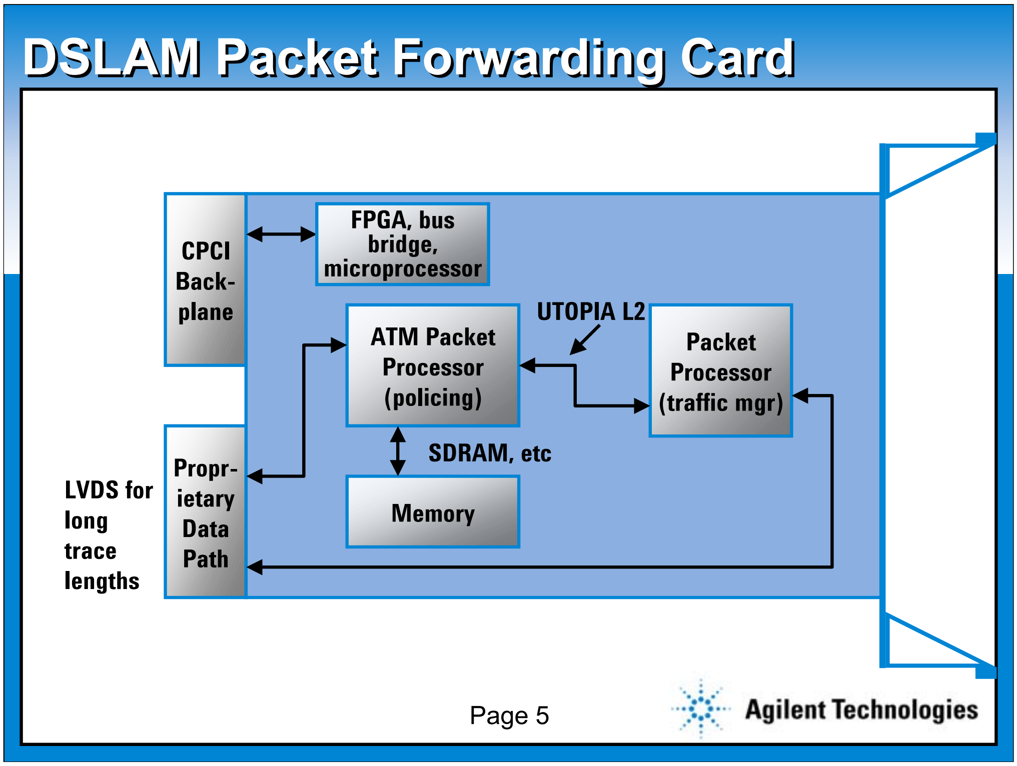
DSLAM (DSL Access Multiplexer) Line Card



This is a rough block diagram of a line card from a DSL Access Multiplexer, or DSLAM. A number of transceivers perform ATM line transmit & receive functions, directing their signals through a multi-port PHY (or physical interface) chip. That chip then routes the ATM cells from all of the transceivers across a UTOPIA bus to packet processing hardware. In this example the packet processing hardware is on the “Core Card”, which is shown in the next slide.

The line card also employs some kind of microprocessor which is used to configure and monitor the behavior of the line card, and communicate with the core card and other system controllers through a standard control bus (in this case Compact PCI).

ATM cell traffic is directed to the packet processing hardware on the core card using serialized LVDS lines.



DSLAM Packet Forwarding Card

CPCI Back-plane

FPGA, bus bridge, microprocessor

ATM Packet Processor (policing)

UTOPIA L2

Packet Processor (traffic mgr)

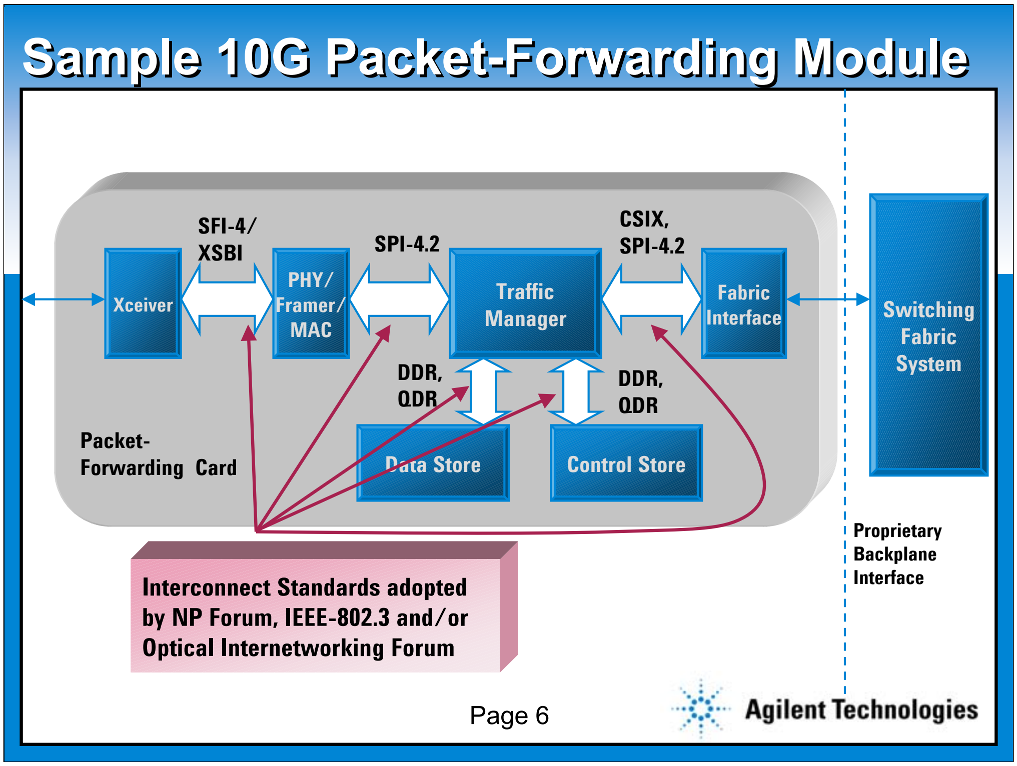
SDRAM, etc

Memory

LVDS for long trace lengths

Propr-ietary Data Path

On the packet processing card of this DSLAM design, packet policing and traffic management functions are performed using commercial chips. Packets/cells are passed around using UTOPIA buses. Policing involves deciding if packets should be forwarded or not. Traffic management involves buffering low-priority traffic while higher-priority packets go through without delay. This allows service providers to create varying levels of service at different pricing levels.



Sample 10G Packet-Forwarding Module

Interconnect Standards adopted by NP Forum, IEEE-802.3 and/or Optical Internetworking Forum

Agilent Technologies

In a core router, there are three main subsystems: line/packet forwarding cards, switching fabric, and system control cards. Packet forwarding cards perform data transmit and receive operations, a portion of the routing table lookup, and policing/scheduling. The switch fabric subsystem contains a high-speed crosspoint switch. System controllers perform routing decisions (that aren't handled locally on the packet-forwarding cards), handle route changes, and perform system management and monitoring operations. The system board typically communicates with the line cards through the control bus, often a Compact PCI interface.

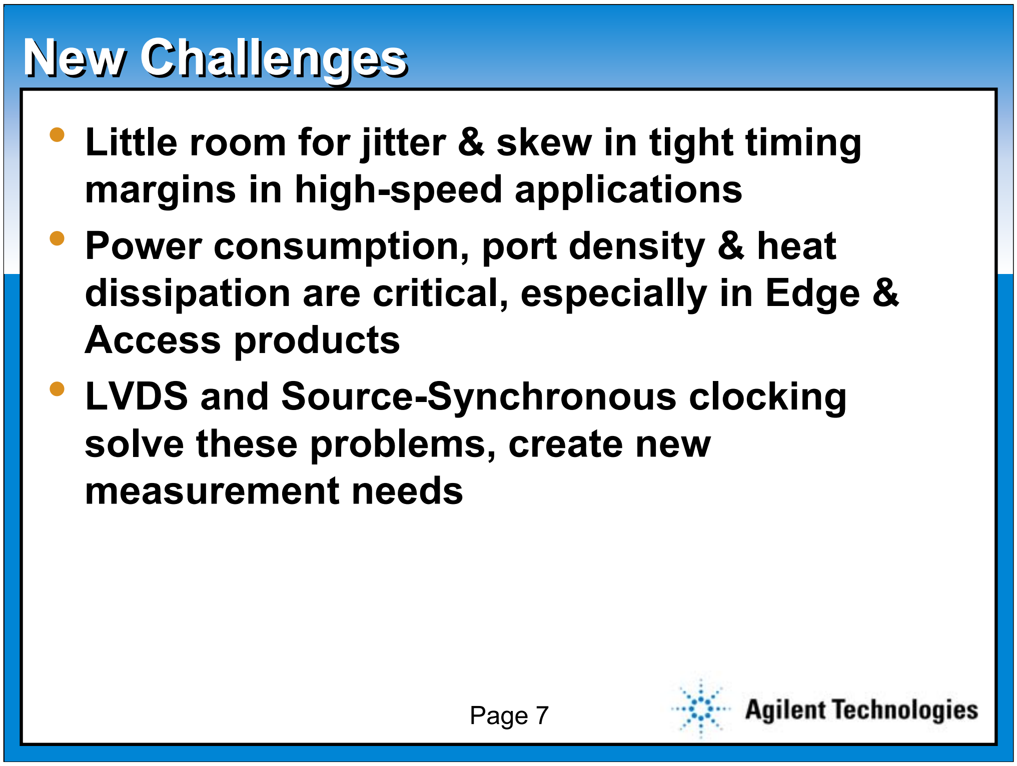
The "traffic manager" can be any combination of FPGAs, ASICs, and programmable Network Processors. This is where route table lookup and priority scheduling occur. If packets need to be buffered to allow higher-priority traffic through, they are buffered here in data storage. The traffic manager accesses a subset of the routing tables in local control storage, which is often SRAM or perhaps a Ternary Content Addressable Memory, or TCAM.

The Network Processing Forum, Optical Internetworking Forum, and IEEE are advocating the use of the following bus standards:

Description	Speed	Forum Name	Bus Standard
Data Path	10 Gb/s	Streaming Interface	SPI-4.2 (aka POS-PHY 4, FlexBus-4 Phase 2)
Switch Fabric Interface	2.5 Gb/s	Fabric Interface	CSIX-L1
Switch Fabric Interface protocol	10 Gb/s	Fabric Interface	Variant of SPI-4.2 carrying CSIX
Control Store Interface 10 Gb/s	Lookaside Interface	Based on QDR SDRAM spec	
SERDES->FRAMER 10 Gb/s	SERDES-Framer Interface		SFI-4, XSBI

While no specific standard was defined for the 2.5 Gb/sec lookaside interface, DDR SDRAM is popular.

Most of these bus standards employ source-synchronous clocking, and either LVDS or HSTL voltage levels. These technologies are discussed in the next slides.



New Challenges

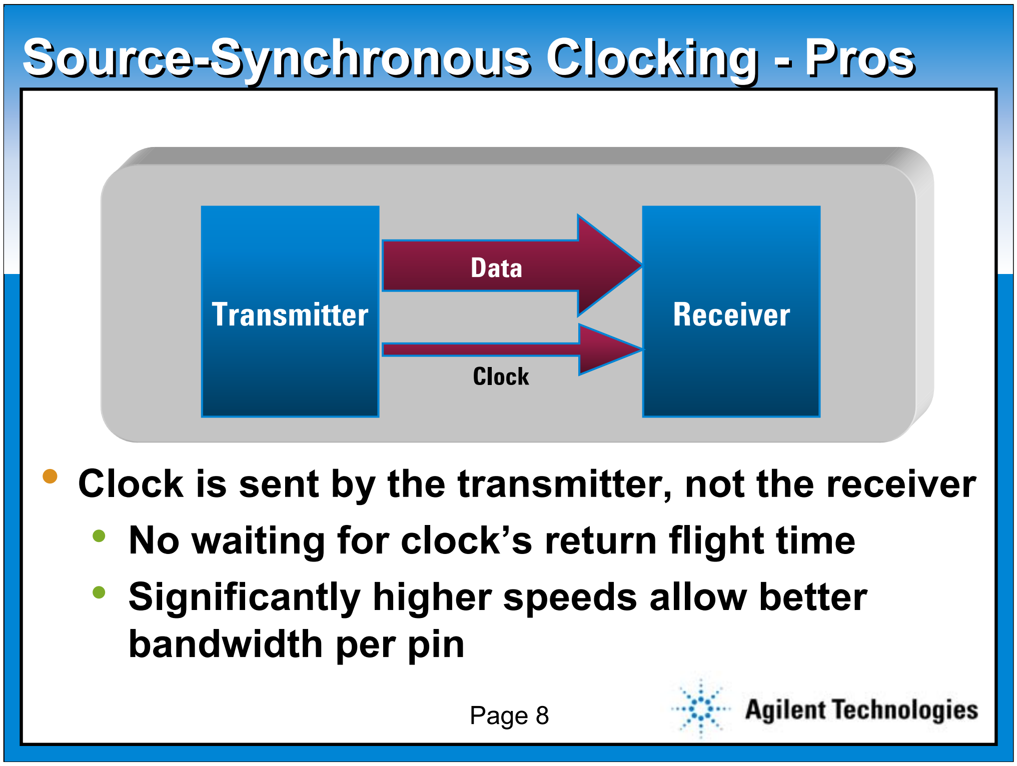
- Little room for jitter & skew in tight timing margins in high-speed applications
- Power consumption, port density & heat dissipation are critical, especially in Edge & Access products
- LVDS and Source-Synchronous clocking solve these problems, create new measurement needs

In order to increase data speeds, timing margins (setup & hold times) have to get smaller. Also control of skew across bus channels needs to be tightly controlled, either through rigid control of circuit board trace lengths, or the use of synchronizing logic in receiving devices. Source-Synchronous clocking (having the transmitting device send the clock instead of the receiver) allows us to dramatically cut setup & hold times, allowing higher bit rates per channel.

Power and heat become critical as well. In X-ray physics, the words “energy” and “frequency” mean the same thing. At constant voltage levels, the same is true in electronics. The higher the frequency, the higher the energy. To combat the rise in power consumption (and therefore heat generation), voltage levels need to decrease as clock rates rise. The rise in popularity of computers and the internet have driven global power consumption to new heights; the supply of electricity isn’t keeping pace with demand, and the impact of increased power generation on the environment is another problem.

Rising bit rates increase RF emissions from and interference to our electronics gear. 2.5 Gbits/sec is close to commonly used cellular telephone frequencies.

Low-Voltage Differential Signaling (LVDS) addresses these problems. The low voltage levels reduce power consumption per unit bandwidth; the differential signaling (using two wires and determining the 1 or 0 by the difference in voltage across the two wires) reduces noise emission and increases the immunity to external noise.

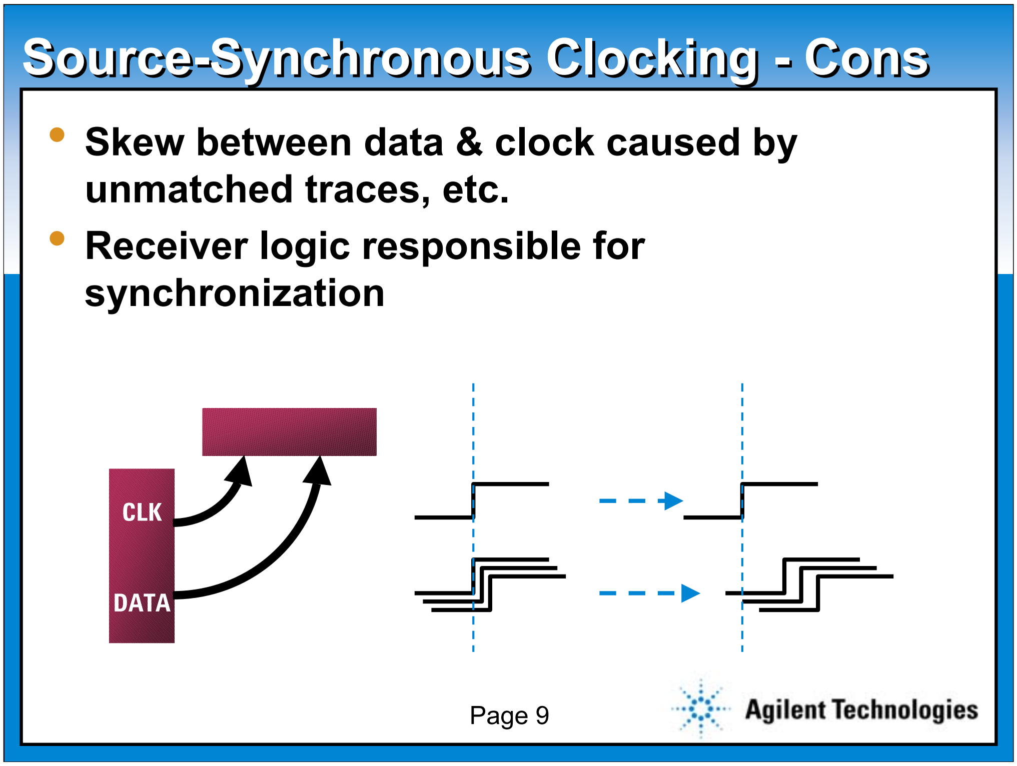


Source-Synchronous Clocking - Pros

- Clock is sent by the transmitter, not the receiver
  - No waiting for clock's return flight time
  - Significantly higher speeds allow better bandwidth per pin

In source-synchronous clocking, the transmitting device sends the clock signal with the data being transmitted. This allows each bus to have its own clock, rather than requiring a tight control over clock distribution across a system, thereby enabling much higher rates.

IO ports on chips can run at large multiples of the internal clock rates, allowing the same amount of data transfer with fewer pins used. Since the gate count in a chip grows much faster than pin count, this is critical.

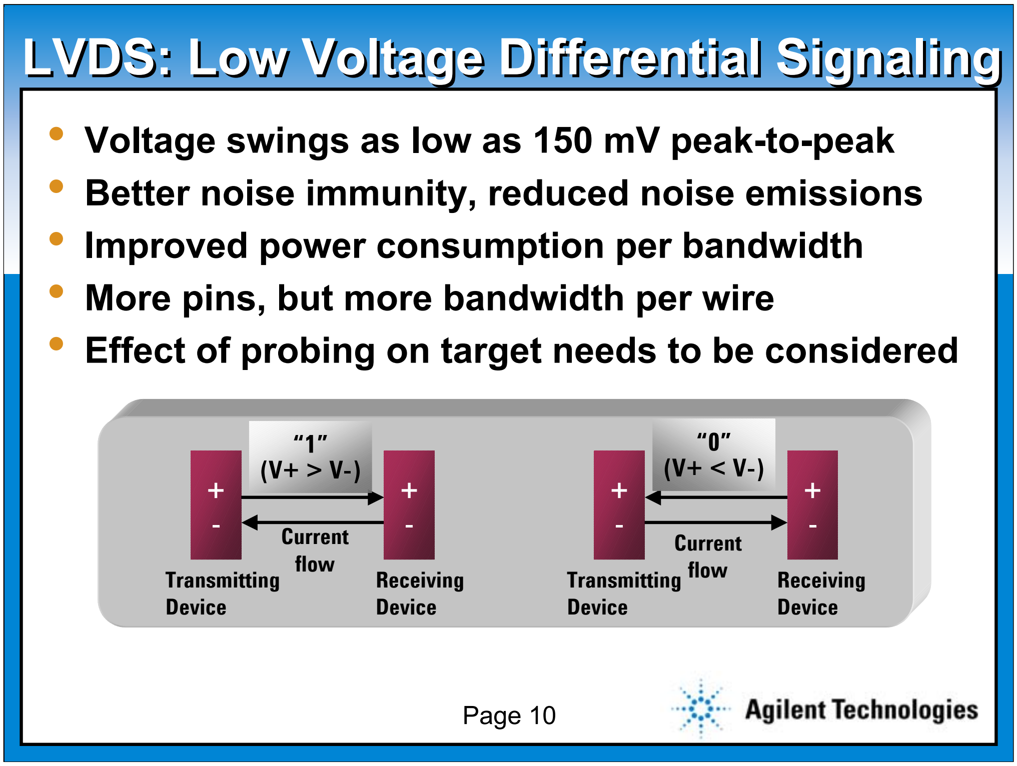


Source-Synchronous Clocking - Cons

- Skew between data & clock caused by unmatched traces, etc.
- Receiver logic responsible for synchronization

Agilent Technologies

Unmatched trace lengths between chips has always created skew; the tighter margins required to run at high speeds make it a more critical issue. In many high-speed bus standards, skew is accepted as inevitable, and the receiving device is required to provide some synchronizing logic. The sending device will periodically transmit a “training sequence”, basically a set of lower-frequency square waves, so that the receiver can compute the delay between edges across the channels and de-skew the bus.



LVDS: Low Voltage Differential Signaling

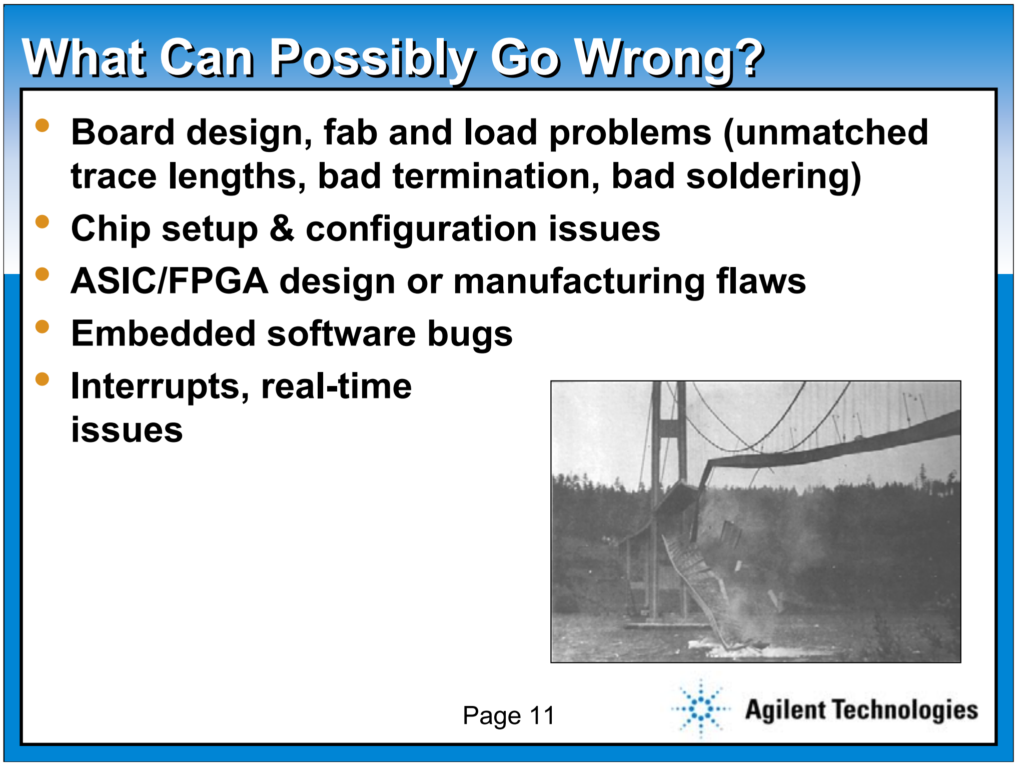
- Voltage swings as low as 150 mV peak-to-peak
- Better noise immunity, reduced noise emissions
- Improved power consumption per bandwidth
- More pins, but more bandwidth per wire
- Effect of probing on target needs to be considered

"1"
(V+ > V-)
+
-
+
-
Current flow
Transmitting Device
Receiving Device
"0"
(V+ < V-)
+
-
+
-
Current flow
Transmitting Device
Receiving Device

Agilent Technologies

As mentioned previously, LVDS brings better power consumption per bandwidth and better noise immunity. However the low voltage swings are below the sensitivity threshold of many commercial logic analyzers, making digital test difficult.

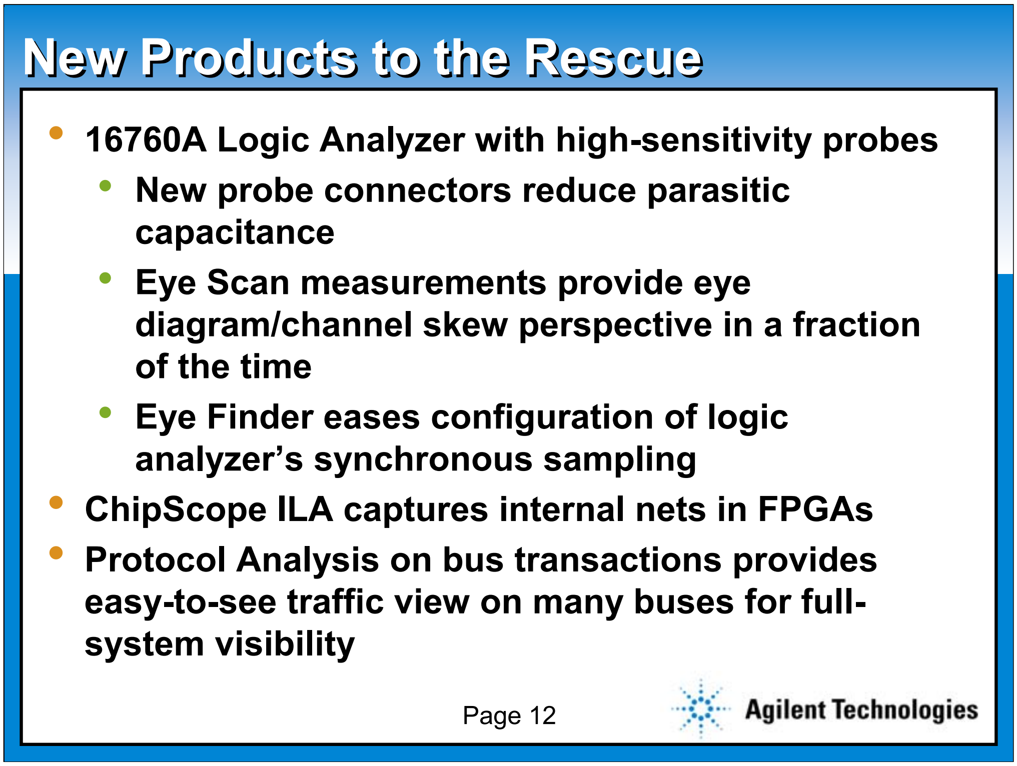
Another problem for test is the move toward point-to-point instead of multi-drop bus terminations. Line drivers generate just enough current for one receiver, and siphoning some off to a passive probe can hurt the performance of the target device. Also, signals are often terminated with resistors contained inside the chips, making the choice of probing location more difficult. Finally the low voltage and high speed aggravate the capacitive loading effects of connecting probes.



What Can Possibly Go Wrong?

- Board design, fab and load problems (unmatched trace lengths, bad termination, bad soldering)
- Chip setup & configuration issues
- ASIC/FPGA design or manufacturing flaws
- Embedded software bugs
- Interrupts, real-time issues

After thinking about the various challenges for test that are created with these new technologies, one might conclude that avoiding test altogether is the best strategy. Simply eliminate all possible flaws before building the first prototype. Unfortunately this is not a good strategy. Simulation can reduce risk at the time of prototype validation, but it can only go so far; to fully simulate a complex system requires immense quantities of compute time, and cannot always be programmed to cover every possibility that can occur in the real world.



New Products to the Rescue

- 16760A Logic Analyzer with high-sensitivity probes
  - New probe connectors reduce parasitic capacitance
  - Eye Scan measurements provide eye diagram/channel skew perspective in a fraction of the time
  - Eye Finder eases configuration of logic analyzer’s synchronous sampling
- ChipScope ILA captures internal nets in FPGAs
- Protocol Analysis on bus transactions provides easy-to-see traffic view on many buses for full-system visibility

Page 12

Agilent Technologies

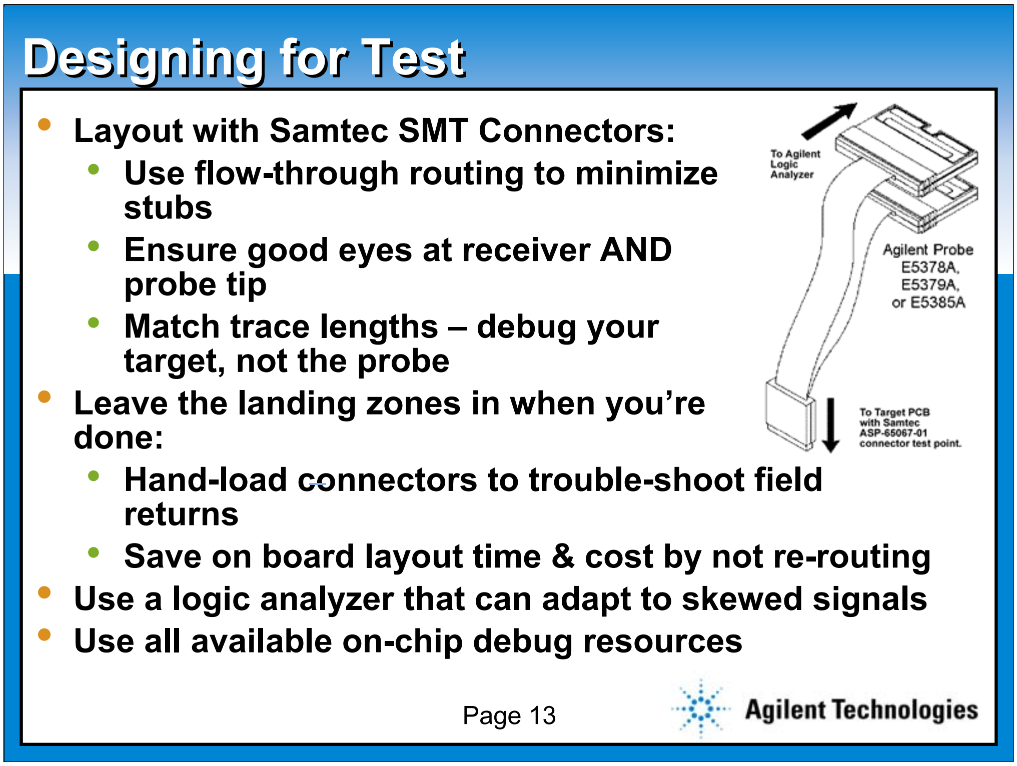
Now that we’ve concluded that success is impossible, let’s see what we can do about it.

To meet these new challenges, Agilent Technologies has developed the 16760A logic analysis module. This new product is capable of synchronous sampling on buses running up to 1.5 Gbits/sec (per channel), and on both single-ended and differential signals. Voltage swings can be as low as 200 mV, and data valid windows can be as small as 500 ps (or even smaller in good conditions). Sampling position and threshold can be adjusted in 10 picosecond and 10 mV steps.

A new surface mount connector (100-pin Samtec ASP-65067-01) and matching termination adapter provide a very low capacitance (1.5 pF) on target signals, allowing successful data capture at speed without significant degradation of the signal bandwidth.

The front end acquisition system in the 16760A is quite like having an oscilloscope sampler on every channel. This new technology has enabled “Eye Scan”, a measurement tool that utilizes the special acquisition system to generate eye diagrams synchronous to the target clock edges on all channels simultaneously. This is not only an effective tool for signal integrity characterization, but a major step in productivity of such measurements.

Finally, new bus analysis tools provide protocol decoding and analysis for many of the standard buses used in IP routers to provide at-a-glance visibility of packet flow



Designing for Test

- Layout with Samtec SMT Connectors:
  - Use flow-through routing to minimize stubs
  - Ensure good eyes at receiver AND probe tip
  - Match trace lengths – debug your target, not the probe
- Leave the landing zones in when you're done:
  - Hand-load connectors to trouble-shoot field returns
  - Save on board layout time & cost by not re-routing
- Use a logic analyzer that can adapt to skewed signals
- Use all available on-chip debug resources

To Agilent Logic Analyzer
Agilent Probe E5378A, E5379A, or E5385A
To Target PCB with Samtec ASP-65067-01 connector test point.

Agilent Technologies

A lot of time at "crunch time" can be saved by planning ahead for test & measurement needs. The following steps can be taken at design time:

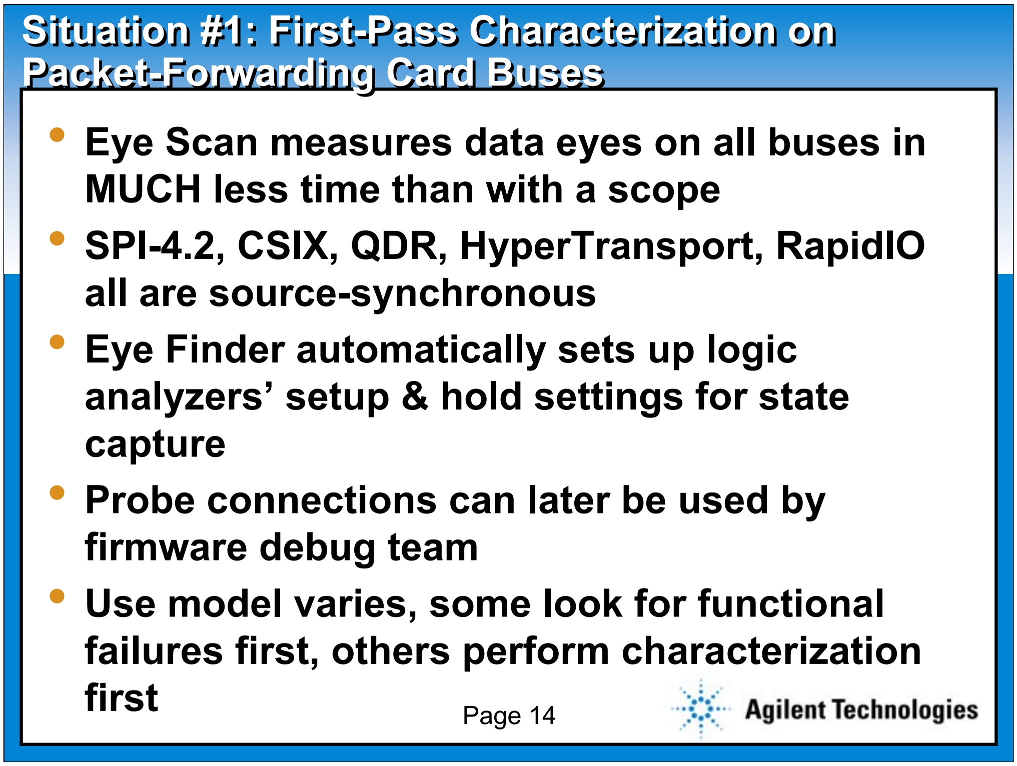
1. Route external buses to test connectors
2. Design in some debug logic in your ASICs or FPGAs
3. Learn how to make use of any on-chip debug resources provided with your commercial chips

The probe connectors will be helpful not only for Eye Scan signal characterization, but in hardware and software troubleshooting. With the right PCB technology the Samtec connector can be routed-through (the traces pass straight through, going between pins) creating very minimal stub effects. In-depth application notes are available from agilent.com to help incorporate the probes.

Many board designers will make a "final cut" of their board once their debug job is finished. It's very important to LEAVE THE LANDING ZONES on the board layout! If your board is a reference design, your customers will appreciate the ability to characterize chips with the probe points. Additionally, if a defect makes it all the way to the production boards, and customers discover them, you'll appreciate being able to debug the problem. You can choose to no-load the connectors (leaving empty pads on the board) if you want to save the cost, then the connectors can be hand-loaded later.

With all the opportunities for data skew, it's important to use a logic analyzer that can accommodate unique sampling positions on each channel. Agilent's Eye Finder makes it a snap to capture skewed signals.

Many chips come with tools for on-chip debug, and not just microprocessors. Xilinx's ChipScope ILA tool can capture logic state data from internal nets inside an FPGA by dropping in some inexpensive IP. The internally captured data can then be exported to the Agilent 167XX system and time correlated to data from external buses.



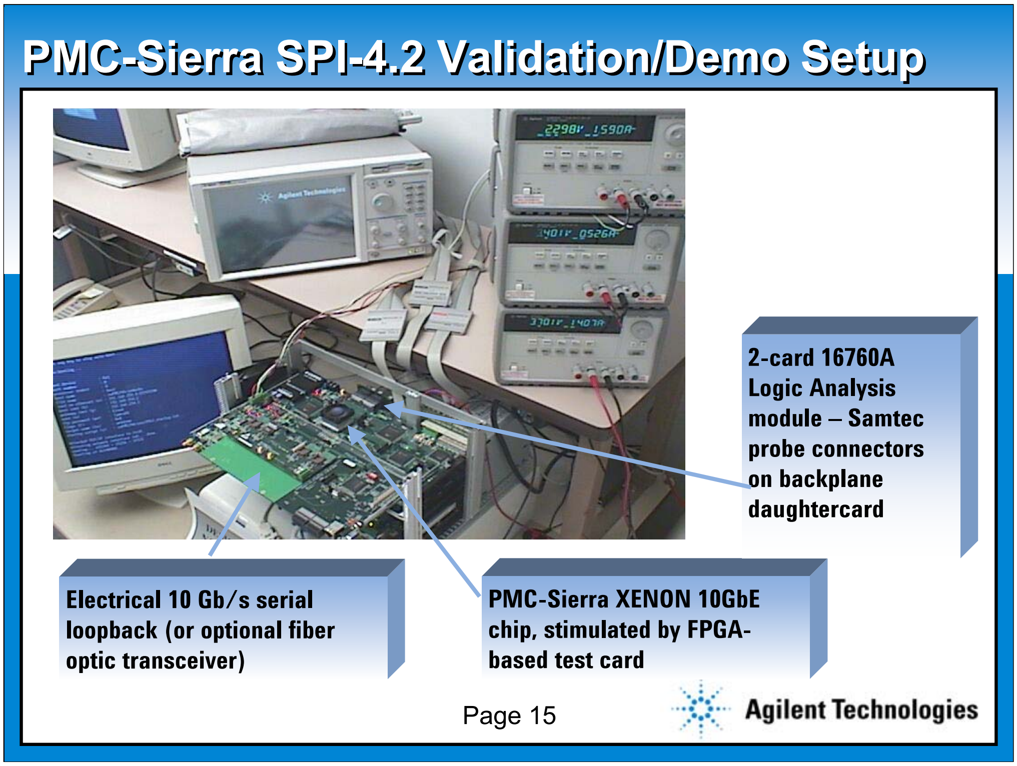
Situation #1: First-Pass Characterization on Packet-Forwarding Card Buses

- Eye Scan measures data eyes on all buses in MUCH less time than with a scope
- SPI-4.2, CSIX, QDR, HyperTransport, RapidIO all are source-synchronous
- Eye Finder automatically sets up logic analyzers’ setup & hold settings for state capture
- Probe connections can later be used by firmware debug team
- Use model varies, some look for functional failures first, others perform characterization first

Our first example scenario is a logical first step in prototype turn-on, characterization of signal integrity. This step is often performed by teams of technicians working around the clock with oscilloscopes and a lot of manual intervention. Simply moving a high-bandwidth scope probe from one signal to the next requires time.

We’ll use the 16760A’s Eye Scan feature to examine the quality of the data eye on a SPI-4.2 (aka POS-PHY Level 4, FlexBus-4 phase 2) bus. This same technique can be applied to other LVDS Source-Synchronous buses such as HyperTransport, CSIX-L1, RapidIO, and DDR and QDR memories.

Once the data eyes are validated to be good, we can use the same 16760A card to capture state data, and decode the bus transactions and packet protocols traveling on the bus, allowing the powerful tools to be shared across characterization and validation teams.



PMC-Sierra SPI-4.2 Validation/Demo Setup

2-card 16760A Logic Analysis module – Samtec probe connectors on backplane daughtercard

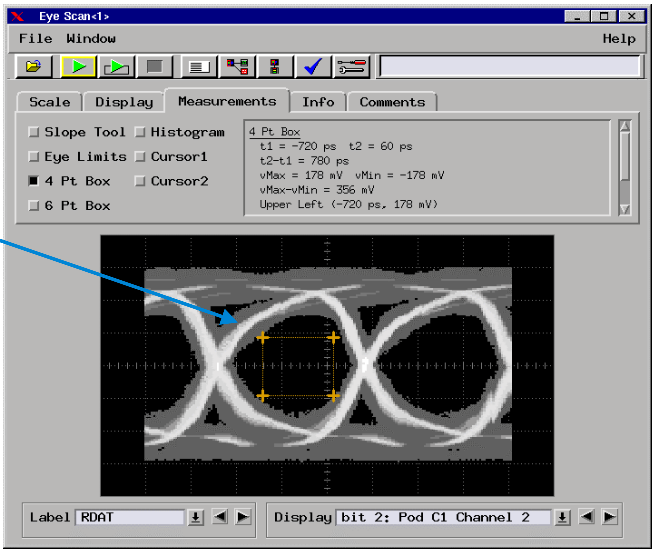
Electrical 10 Gb/s serial loopback (or optional fiber optic transceiver)

PMC-Sierra XENON 10GbE chip, stimulated by FPGA-based test card

Here is a picture of the setup we used for this scenario. This is a demonstration system from PMC-Sierra, showing their XENON chip for OC-192 Packet over SONET, ATM, and 10G Ethernet. A pair of Xilinx Virtex-II FPGAs are used for traffic generation on the SPI-4.2 side, and 10 Gb/sec serial data is looped back on an electrical loopback daughter card. The electrical loopback can be replaced with an optical transceiver module for connection to other devices.

A 2-card set of 16760A modules is connected to the SPI-4.2 buses (transmit and receive directions, each with 16 data channels, a control bit, and clock). The test system allows users to configure the clock speed, but in this example the clock is running at roughly 311 MHz, for a data rate of 622 Mbits/sec per channel.

SPI-4.2 Eye Scan – Single Data Channel



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

```
4 Pt Box
  t1 = -720 ps  t2 = 60 ps
  t2-t1 = 780 ps
  vMax = 178 mV  vMin = -178 mV
  vMax-vMin = 356 mV
  Upper Left (-720 ps, 178 mV)
```

Label RDAT

Display bit 2: Pod C1 Channel 2

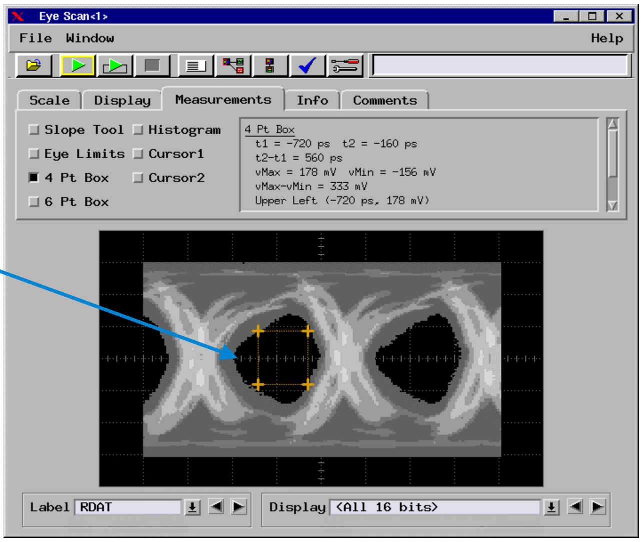
Data Valid Window 780 ps x 356 mV

Here is an Eye Scan of a single channel on the SPI-4.2 bus coming out of the XENON chip. We have used a 4-point box measurement tool to measure the data valid window, which is 780 picoseconds wide and 356 millivolts high. There are a number of measurements that can be easily made from the Eye Scan display, including 4- and 6-point boxes, slope, and histograms.

The Eye data can be shown in greyscale, or color, on user-adjustable linear and log scales.

SPI-4.2 Eye Scan – Composite of 16 Data Bits

Data Valid Window 560 ps x 333 mV across ALL 16 CHANNELS



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

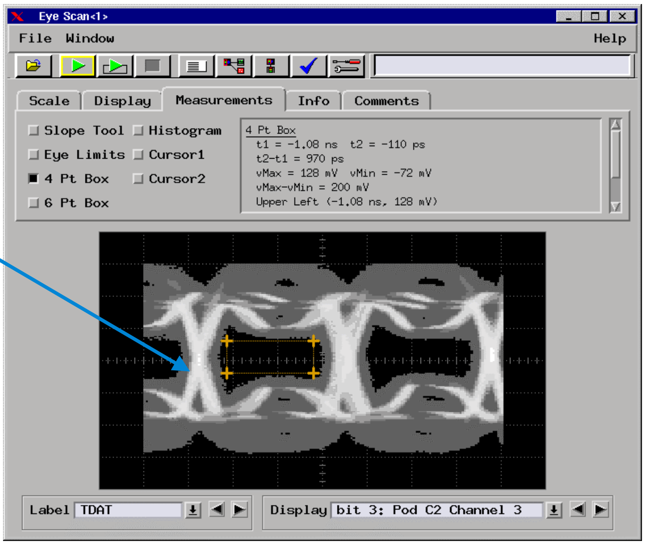
```
4 Pt Box
  t1 = -720 ps  t2 = -160 ps
  t2-t1 = 560 ps
  vMax = 178 mV  vMin = -156 mV
  vMax-vMin = 333 mV
  Upper Left (-720 ps, 178 mV)
```

Now let’s overlay all 16 channels of the SPI-4.2 data bus. Notice now that the composite eye is smaller, mainly due to skew between channels. This is less critical to a SPI-4.2 user because training patterns and receiver synchronization logic makes up for skew, but it can give you an idea of the overall quality of the interconnect.

Additionally you can choose to display a single channel highlighted over the composite display, allowing you to quickly pinpoint which channels are misbehaving.

SPI-4.2 Eye Scan – Reflection Problems

Data Valid Window
970 ps x 200 mV



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

```
4 Pt Box
  t1 = -1.08 ns  t2 = -110 ps
  t2-t1 = 970 ps
  vMax = 128 mV  vMin = -72 mV
  vMax-vMin = 200 mV
  Upper Left (-1.08 ns, 128 mV)
```

Label TDAT

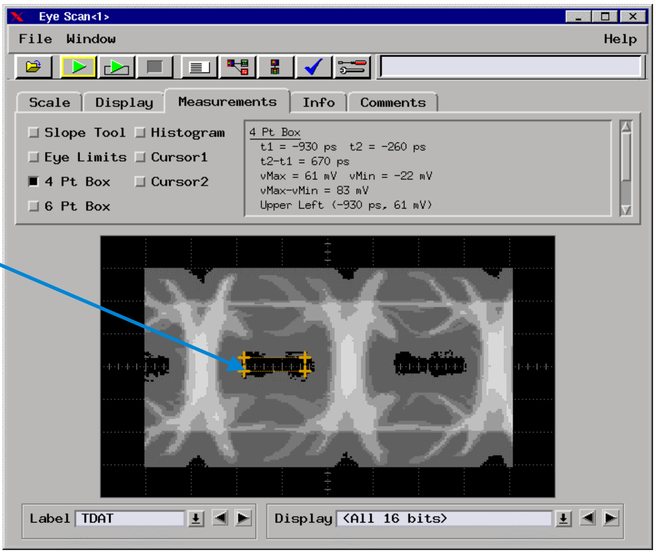
Display bit 3: Pod C2 Channel 3

Now here is a very different picture. This eye has a much smaller data valid window, only 200 millivolts high. It appears that some improper termination or interconnect has caused some reflections. Even so, the eye is big enough for successful data capture in the 16760A, and for adequate data transmission from sender to receiver.

This data was captured by measuring the eye of the “transmit” bus on the PHY chip (coming from an FPGA on a separate card, traveling across a backplane connector) relative to the clock generated by the “receive” bus of the PHY chip, so it’s really not a fair measurement. It did allow us to make an interesting eye however.

The Eye Finder capability allows us to use either of the two clocks (transmit and receive) for our measurements across both buses simply by adjusting the sampling positions to the center of the stable windows.

SPI-4.2 Eye Scan – Composite



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

```
4 Pt Box
  t1 = -930 ps  t2 = -260 ps
  t2-t1 = 670 ps
  vMax = 61 mV  vMin = -22 mV
  vMax-vMin = 83 mV
  Upper Left (-930 ps, 61 mV)
```

Label TDAT

Display <All 16 bits>

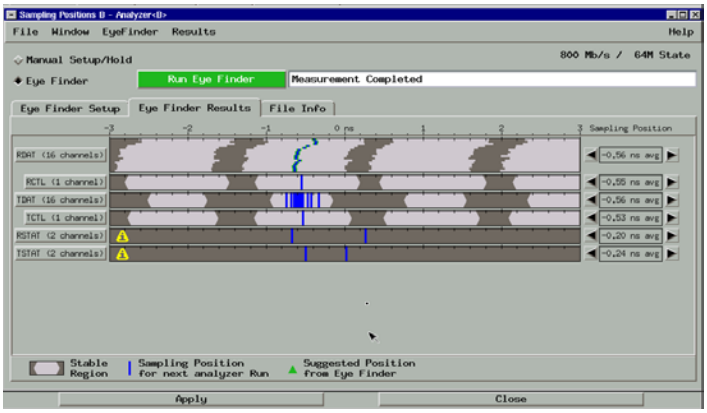
Data Valid Window 670 ps x 83 mV

Composite Eye is less important with SPI-4.2, training patterns & sync logic correct for skew

Finally here is a composite of the bad eyes from all 16 channels on the data bus. Just to show that we can!

Eye Finder – Individual Sampling Position Per Bit

Possible sources of skew in this measurement include stubs to probe connector; matched trace lengths to the probe let you see the skews on your board



800 Mb/s / 64M State

○ Manual Setup/Hold

● Eye Finder

Measurement Completed

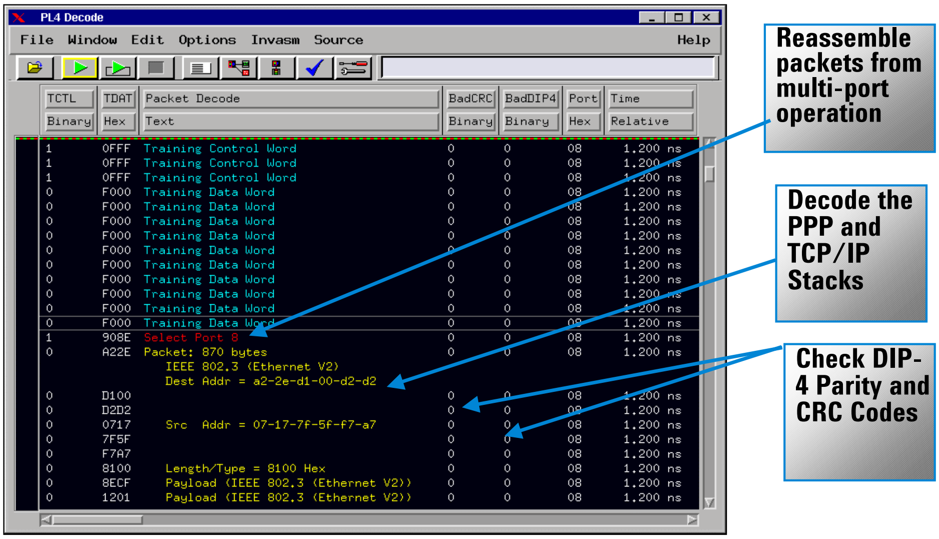
	Sampling Position
RDAT (16 channels)	-0.56 ns avg
RCTL (1 channel)	-0.55 ns avg
TDAT (16 channels)	-0.56 ns avg
TCTL (1 channel)	-0.53 ns avg
RSTAT (2 channels)	-0.20 ns avg
TSTAT (2 channels)	-0.24 ns avg

Once the eyes are determined to be good, we're interested in running something other than PRBS (Pseudorandom Byte Sequence) data and analyze the bus cycles and packet traffic. To prepare for these State Mode measurements, we use Eye Finder to choose the optimal sampling location on each channel.

As you can see, the channels on the data bus drift from one side to the other. Clearly some traces are not perfectly matched. We'll need to be careful about what we see, however; the mismatch could be on the stubs leading to the probe connector, and not necessarily between the two chips. If you're careful about probing you can use this display as an indication of skew on the board.

The software has automatically found the center of the stable regions (shown by a green mark) relative to the clock edge (located at the 0 ps mark), and automatically configured the logic analyzer to sample at the center of that stable region. If your signals are skewed more than a complete bit-time, you can manually adjust these settings simply by dragging the blue line which indicates the analyzer's sampling position.

State Mode Analysis & Decode – SPI-4.2



```
TCTL    TDAT  Packet Decode                          BadCRC  BadDIP4  Port  Time
Binary  Hex   Text                                   Binary  Binary   Hex   Relative
1       0FFF  Training Control Word                  0       0        08    1.200 ns
1       0FFF  Training Control Word                  0       0        08    1.200 ns
1       0FFF  Training Control Word                  0       0        08    1.200 ns
0       F000  Training Data Word                     0       0        08    1.200 ns
0       F000  Training Data Word                     0       0        08    1.200 ns
0       F000  Training Data Word                     0       0        08    1.200 ns
0       F000  Training Data Word                     0       0        08    1.200 ns
0       F000  Training Data Word                     0       0        08    1.200 ns
0       F000  Training Data Word                     0       0        08    1.200 ns
0       F000  Training Data Word                     0       0        08    1.200 ns
0       F000  Training Data Word                     0       0        08    1.200 ns
0       F000  Training Data Word                     0       0        08    1.200 ns
0       F000  Training Data Word                     0       0        08    1.200 ns
1       908E  Select Port 8                          0       0        08    1.200 ns
0       A22E  Packet: 870 bytes                      0       0        08    1.200 ns
                 IEEE 802.3 (Ethernet V2)
                 Dest Addr = a2-2e-d1-00-d2-d2
0       D100                                         0       0        08    1.200 ns
0       D2D2                                         0       0        08    1.200 ns
0       0717     Src  Addr = 07-17-7f-5f-f7-a7       0       0        08    1.200 ns
0       7F5F                                         0       0        08    1.200 ns
0       F7A7                                         0       0        08    1.200 ns
0       8100     Length/Type = 8100 Hex              0       0        08    1.200 ns
0       8ECF     Payload (IEEE 802.3 (Ethernet V2))  0       0        08    1.200 ns
0       1201     Payload (IEEE 802.3 (Ethernet V2))  0       0        08    1.200 ns
```

Reassemble packets from multi-port operation

Decode the PPP and TCP/IP Stacks

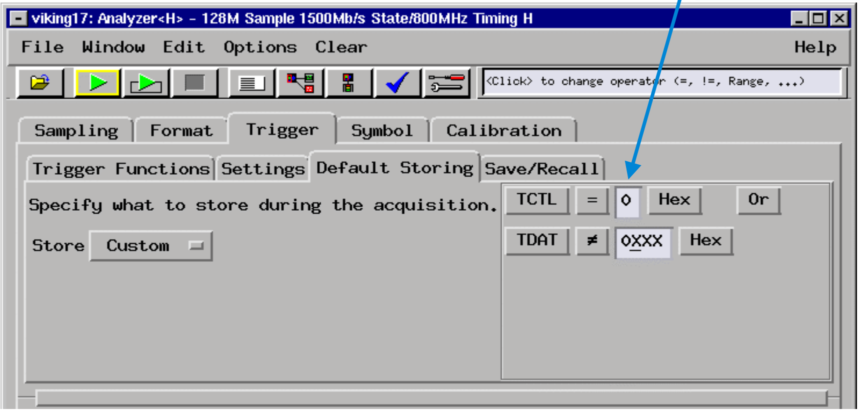
Check DIP-4 Parity and CRC Codes

Once we're operating in state mode we're ready to start looking at bus traffic from a higher perspective. The POS-PHY Level 4 toolset converts state mode SPI-4.2 data into decoded text, labelling Idle states, training patterns, port selection and other control states, and performs multi-layer protocol decode on packet data.

Postprocessing software computes DIP-4 parity and CRC checksums (when applicable) and compares the values to those transmitted on the bus. You can then use the Listing display's search capabilities to look for data corruption, searching for BadDIP-4 = 1 for example.

Store Qualification of Idle States

Choose not to save Idle states in trace buffer



Specify what to store during the acquisition.

Store Custom

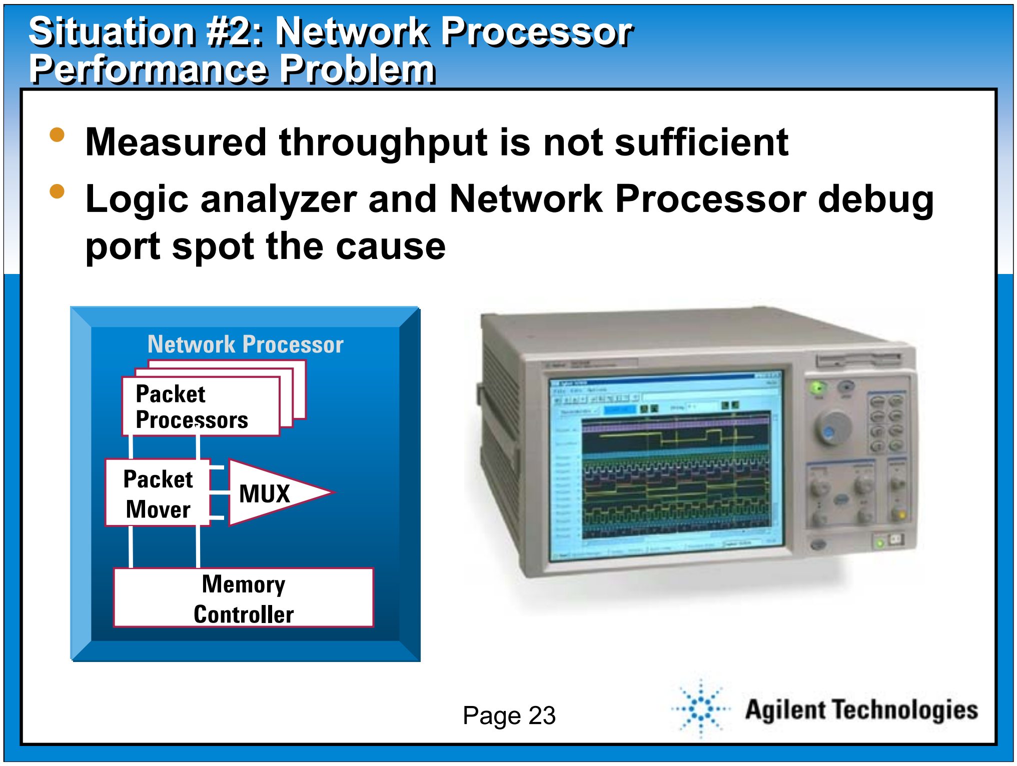
TCTL = 0 Hex Or

TDAT ≠ 0XXX Hex

Agilent Technologies

Another key aspect of state mode analysis is ignoring Idle states on the bus, maximizing the value of the deep acquisition memory. By configuring the 16760A's "Default Storing" as shown, we store all states except those idle states which have no Start of Packet or End of Packet information.

We are now configured and ready to begin performing system-level debug.



Situation #2: Network Processor Performance Problem

- Measured throughput is not sufficient
- Logic analyzer and Network Processor debug port spot the cause

Network Processor

Packet Processors

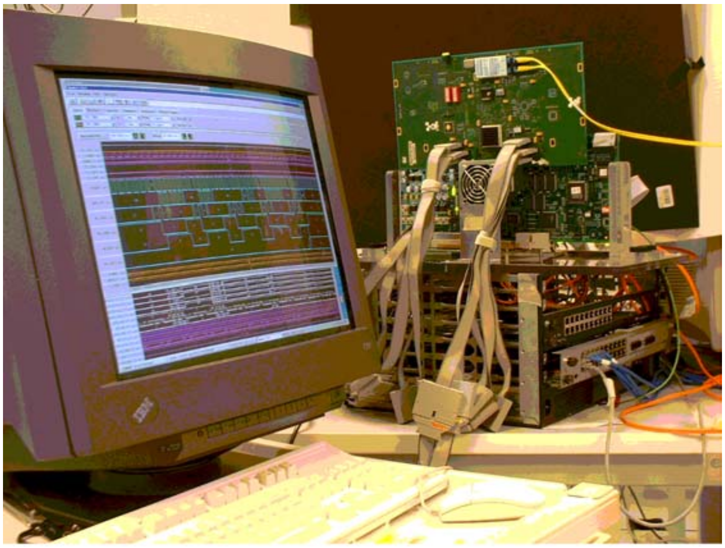
Packet Mover

MUX

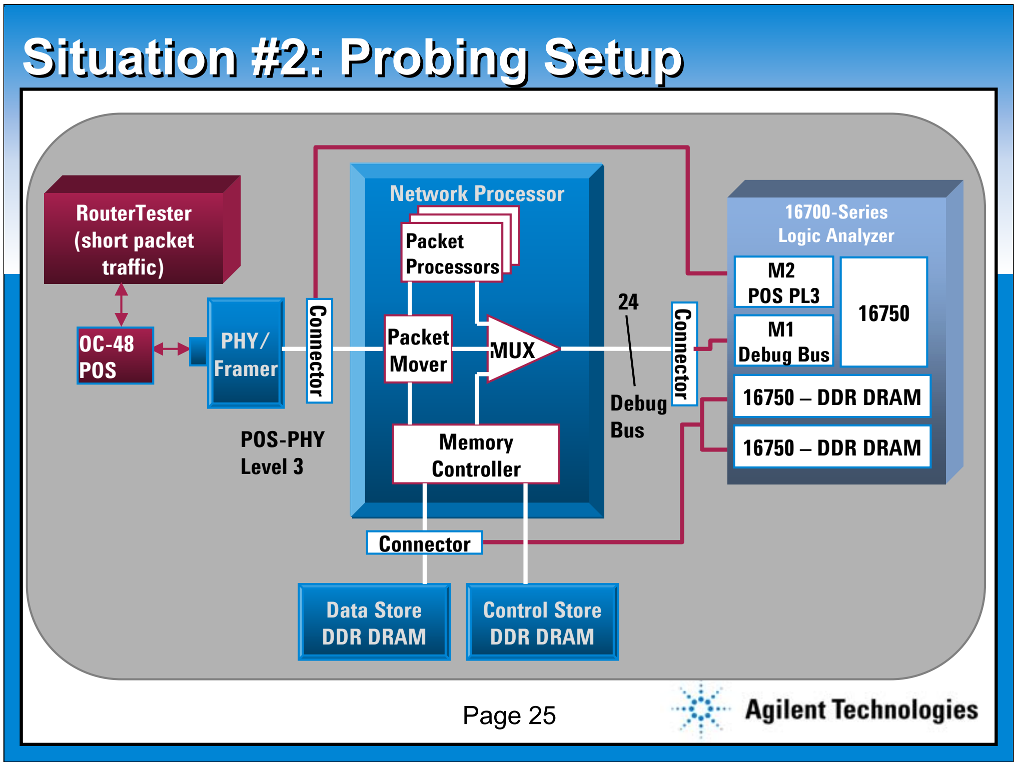
Memory Controller

Our second scenario comes from a customer who was debugging a reference design for a network processor. It was found that packet forwarding performance was not as good as expected, and an Agilent logic analyzer combined with the on-chip debug resources of the NP managed to spot the problem, allowing the engineers to solve it.

IBM Network Processor Validation Platform



Here is a picture of the setup we used for this scenario. This is a reference design incorporating the IBM Network Processor. The network processor is capable of processing packets at OC-48 wire speed. In this case the OC-48 bandwidth is divided into 4 OC-12 quadrants, and two of those quadrants are using 4 OC-3 channels each, showing how bandwidth can be partitioned across channels. The data path interface from the network processor is an AMCC FlexBus, which is very much like a POS-PHY bus except that it can be partitioned into quadrants where each channel uses 8 bits of the 32-bit data bus, and there are 4 sets of control signals (start of packet, data valid, end of packet).



Situation #2: Probing Setup

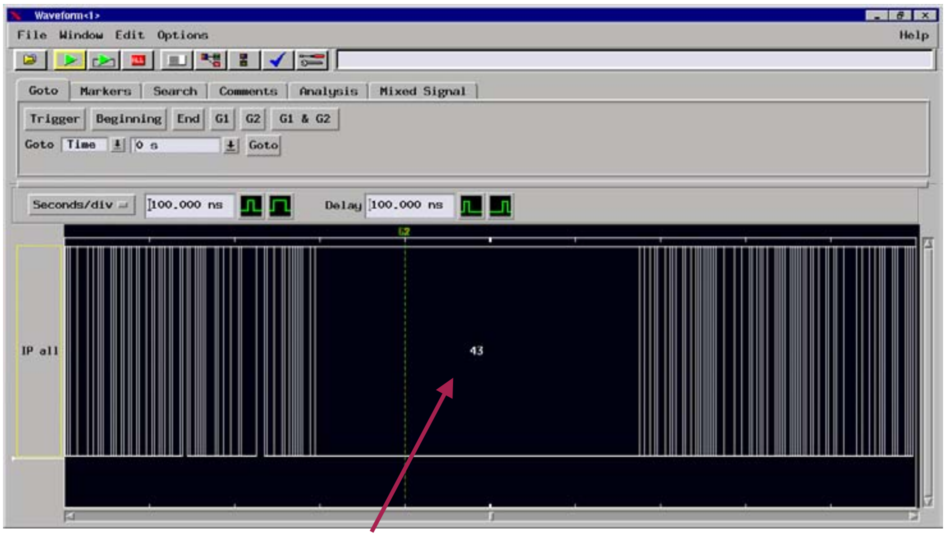
Agilent Technologies

In this network processor, there is a bank of packet processors. When a packet arrives it is dispatched to one of the packet processors, which then performs all policing and route lookup operations, finally shipping that packet to its next destination.

The user was able to route the "instruction pointer" or "program counter" from one of the packet processors to a debug port on the network processor. By looking at the behavior of the instruction pointer, we might determine if any odd behavior is happening in the packet processors.

While one logic analyzer module is connected to the debug port of the NP, other modules probe the PHY bus (POS-PHY Level 3 in this case) and the data store memory (DDR memory). Packet traffic for the measurement is provided by an Agilent Technologies OC-48 RouterTester.

Situation #2: Early Analysis



Goto Time 0 s

Seconds/div 100.000 ns

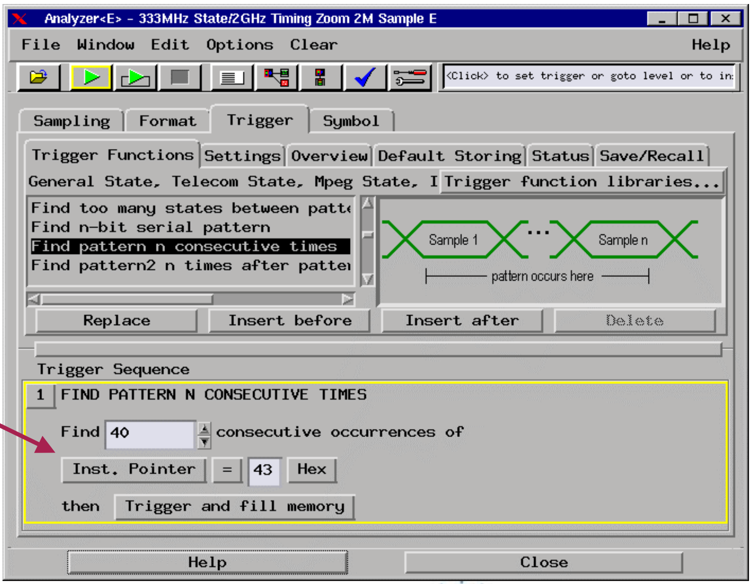
Delay 100.000 ns

Instruction Pointer stuck, packet processor sitting idle

First, we take a cursory look at the behavior of the packet processor Instruction Pointer. By triggering automatically (or trigger on “anything”), and setting up a repetitive run, an occasional stuck value would appear. The user noticed the value was always the same, pointing to a “Dispatch Frame” instruction. For some reason, the frame is stuck waiting to get out of the packet processor and on its way through the system.

Situation #2: Trigger Setup

Trigger when Inst Pointer = 0x43 for 40 consecutive clocks


Analyzer<E> - 333MHz State/2GHz Timing Zoom 2M Sample E
File Window Edit Options Clear Help
<Click> to set trigger or goto level or to in:
Sampling Format Trigger Symbol
Trigger Functions Settings Overview Default Storing Status Save/Recall
General State, Telecom State, Mpeg State, I Trigger function libraries...
Find too many states between patt
Find n-bit serial pattern
Find pattern n consecutive times
Find pattern2 n times after patte
Sample 1
Sample n
pattern occurs here
Replace
Insert before
Insert after
Delete
Help
Close

Trigger Sequence

1 FIND PATTERN N CONSECUTIVE TIMES

Find 40 consecutive occurrences of

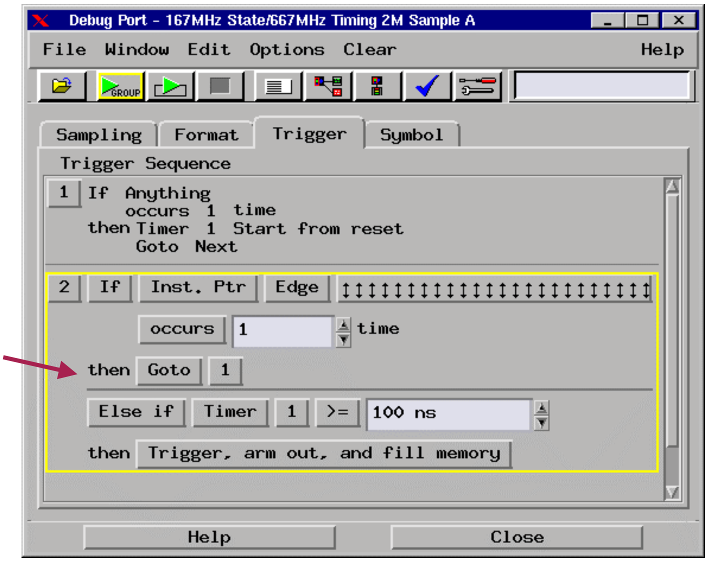
Inst. Pointer = 43 Hex

then Trigger and fill memory

Agilent Technologies

Now that we have narrowed down a symptom of the problem, we need to zero in on the rest of the system at the time of that stuck instruction pointer. There are a few ways to trigger the logic analyzer on the stuck IP. The easiest is shown above, simply selecting to trigger when IP = 0x43 for 40 clocks or more. We can then use that trigger source to look at other buses in the system at the same time to see what’s going on elsewhere.

Situation #2: Alternate Trigger Setup



```
Trigger Sequence
1  If  Anything
       occurs  1  time
   then Timer  1  Start from reset
        Goto  Next

2  If  Inst. Ptr  Edge  ↕↕↕↕↕↕↕↕↕↕↕↕↕↕↕↕↕↕↕↕↕↕↕↕
       occurs  1  time
   then  Goto  1

   Else if  Timer  1  >=  100 ns
   then  Trigger, arm out, and fill memory
```

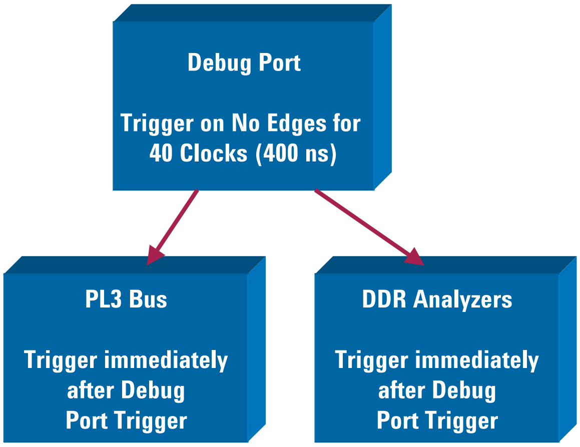
Trigger when no edges occur on the Debug port for 100 ns

Will trigger on ANY stuck instruction, not just a specific one

Agilent Technologies

An alternative way to trigger the analyzer is to trigger when no edges occur on any bit of the bus for 100 nanoseconds. This would allow us to trigger on ANY instruction pointer value that's stuck, without necessarily having to know which one is the culprit.

Situation #2: Trigger Setup



Debug Port

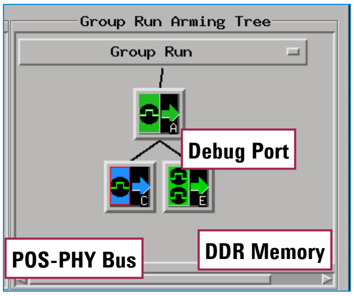
Trigger on No Edges for 40 Clocks (400 ns)

PL3 Bus

Trigger immediately after Debug Port Trigger

DDR Analyzers

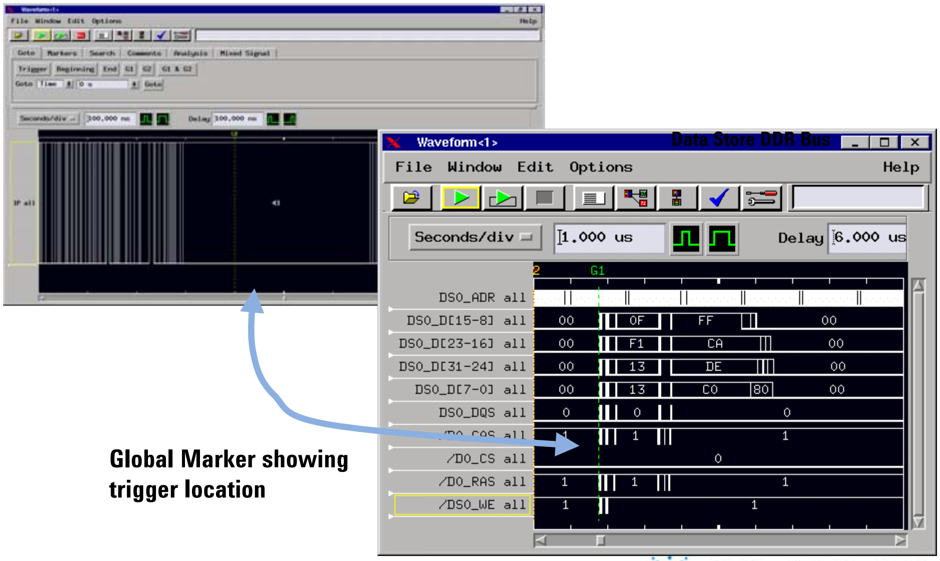
Trigger immediately after Debug Port Trigger



By using the “Intermodule” triggering capabilities of the 167XX system, we export the trigger from the NP’s debug port to the analyzers probing the PHY bus and the Data Store memory. Those analyzers are configured to trigger as soon as they receive the signal from the debug port.

Situation #2: Discovery

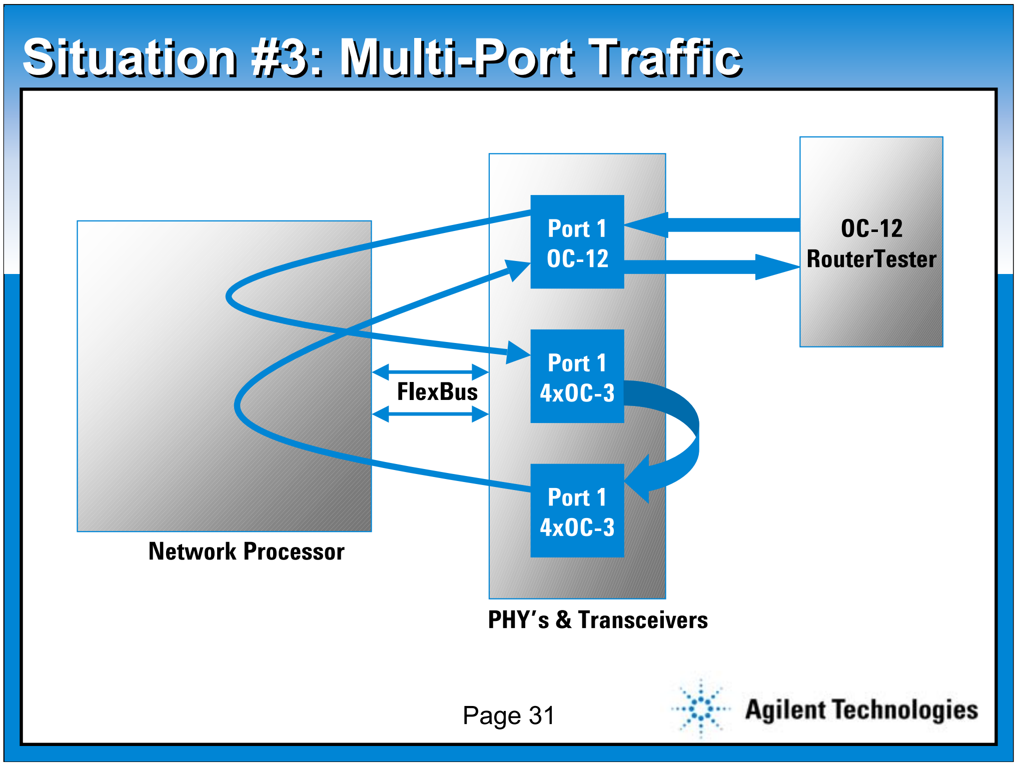
Debug Port (Instruction Pointer)



Global Marker showing trigger location

Agilent Technologies

With our new measurement, we discover that the data store (DDR memory) is busy at the time of the "dispatch frame" instruction that got stuck; the packet processor missed its data window and has to wait for its turn to come around again. The microcode designers now know that they have a priority & scheduling issue, and can retune their scheduling algorithms to improve performance.



Situation #3: Multi-Port Traffic

Port 1
OC-12

OC-12
RouterTester

Port 1
4xOC-3

FlexBus

Port 1
4xOC-3

Network Processor

PHY's & Transceivers

Agilent Technologies

Now let's examine some lower-speed Packet Over SONET (POS) traffic going through the same OC-48 network processor. In this case the bandwidth is split into 4 OC-12 quadrants, and two of the quadrants have 4 OC-3 ports each.

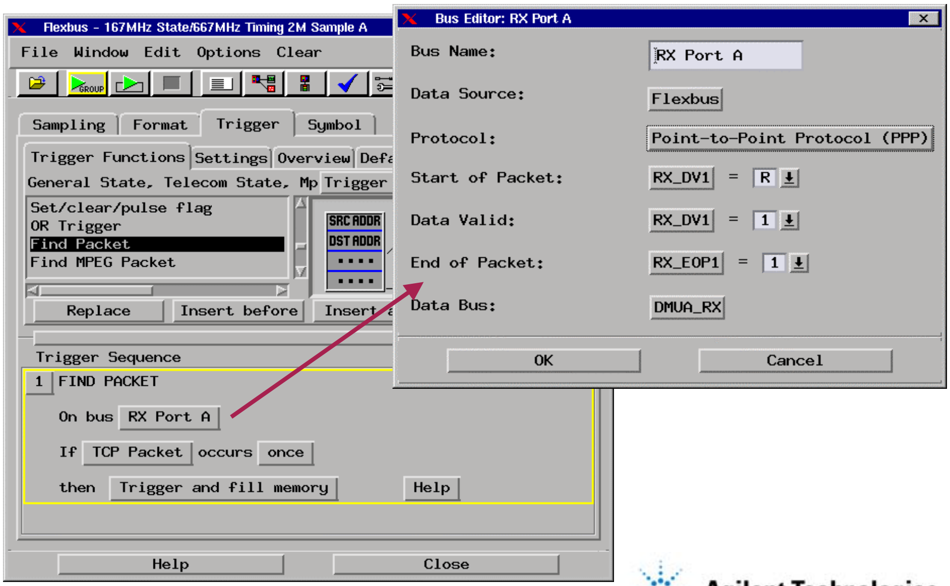
The engineer has routed traffic stimulus from a RouterTester into the OC-12 PORT A receive port. From there the network processor retransmits the packets on PORT B, a 4xOC-3 port. Using 4 loopback cables, that data is received again by PORT C, also a 4xOC-3 port. Finally the packets are retransmitted back out the PORT A transmit interface and back into the RouterTester.

The traffic from the four PHY ports to the network processor travels across an AMCC "FlexBus", which is very similar to POS-PHY and UTOPIA. The bus is configured whereby the 32-bit data bus is segmented into 4 8-bit buses. Each port has its own control signals (start of packet, end of packet, data valid, etc), so we can treat the single FlexBus as 4 POS-PHY or UTOPIA buses each carrying Point-To-Point protocol.

This exercise was done to validate the 4xOC-12 operation of the network processor, but you can imagine similar applications in system-level validation of a router. The RouterTester can provide compliance and performance measurements, while the logic analyzer connections help debug timing, control and data flow issues at the subsystem and component level.

If your RouterTester shows inadequate performance, the timing information from a logic analyzer can immediately pinpoint the bottlenecks.

Multi-port POS FlexBus Trigger


Flexbus - 167MHz State/667MHz Timing 2M Sample A
File Window Edit Options Clear
Sampling Format Trigger Symbol
Trigger Functions Settings Overview Defa
General State, Telecom State, Mp Trigger
Set/clear/pulse flag
OR Trigger
Find Packet
Find MPEG Packet
SRC ADDR
DST ADDR
Replace Insert before Insert a
Help Close
Bus Editor: RX Port A
OK Cancel

Trigger Sequence

1 FIND PACKET

On bus RX Port A

If TCP Packet occurs once

then Trigger and fill memory

Help

Bus Name: RX Port A

Data Source: Flexbus

Protocol: Point-to-Point Protocol (PPP)

Start of Packet: RX_DV1 = R

Data Valid: RX_DV1 = 1

End of Packet: RX_EOP1 = 1

Data Bus: DMUA_RX

Agilent Technologies

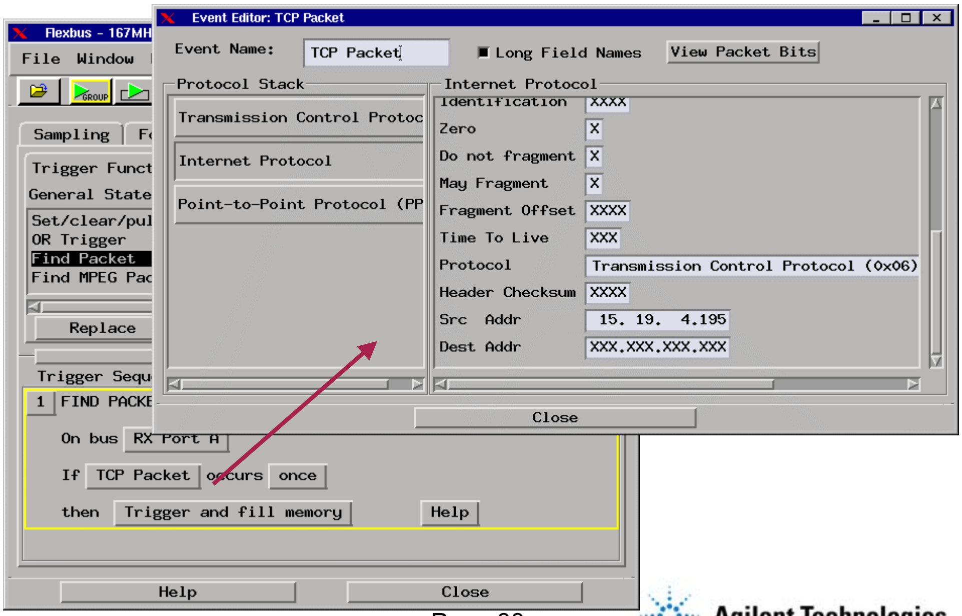
In this measurement, perhaps we notice that certain kinds of packets take longer than others to propagate through the system. For that reason we may want to use the Packet Trigger macro to capture a specific type of packet.

In the Find Packet macro, we define a bus (shown above), providing the following:

- Name of the bus (we have 4 we're interested in in this example)
- Protocol running on the bus (Packet Over SONET uses the Point-to-Point protocol)
- What signals the Start of a packet (in this case the RX_DV1 signal rising is our indication)
- What signals valid data (this is used for filtering Idle states in the decode tools later)
- What signals the End of Packet (again used for decoding)
- What is the label name of the Data bus

If the data bus is very wide, multiple labels can be combined (for a 64-bit CSIX bus, for example)

Multi-port POS FlexBus Trigger



Event Name: TCP Packet

■ Long Field Names

Protocol Stack
- Transmission Control Protoc
- Internet Protocol
- Point-to-Point Protocol (PP

Internet Protocol

Identification	XXXX
Zero	X
Do not fragment	X
May Fragment	X
Fragment Offset	XXXX
Time To Live	XXX
Protocol	Transmission Control Protocol (0x06)
Header Checksum	XXXX
Src Addr	15. 19. 4.195
Dest Addr	XXX.XXX.XXX.XXX

Trigger Funct

General State
- Set/clear/pul
- OR Trigger
- Find Packet
- Find MPEG Pac

Trigger Sequ

1 FIND PACKE

On bus RX Port A

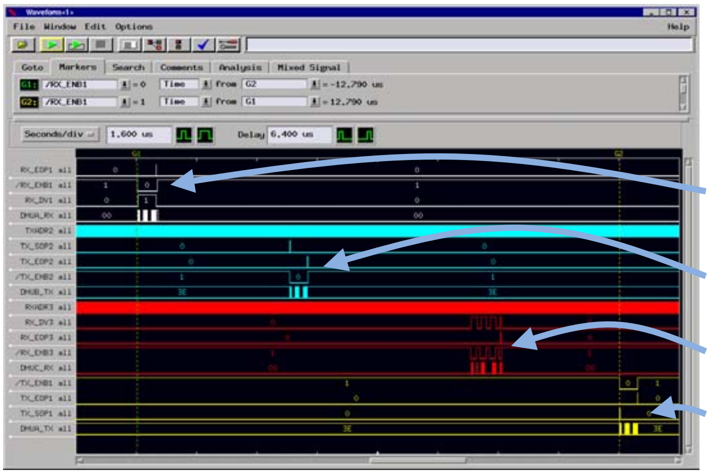
If TCP Packet occurs once

then Trigger and fill memory

Now let’s say we want to trigger specifically on a TCP (Transmission Control Protocol) packet coming from the IP address 15.19.4.195. We can define the trigger event by protocol types and field values. We can edit field values at any level in the stack by selecting the appropriate layer on the left, and entering field values on the right. A value of “X” means don’t-care.

This packet event is then compiled into a trigger sequence that matches the bus width; the same event can be used to trigger on an 8-bit bus, and a 32-bit bus. You don’t have to redefine the same event over and over, the macro recompiles it for each bus.

Multi-port POS FlexBus View



G1: /RX_ENB1 = 0 Time from G2 = -12.790 us

G2: /RX_ENB1 = 1 Time from G1 = 12.790 us

Seconds/div 1.600 us Delay 6.400 us

Packet arrives on OC-12 port 1 RX from Router Tester

Packet re-sent across channelized quad OC-3 port 2

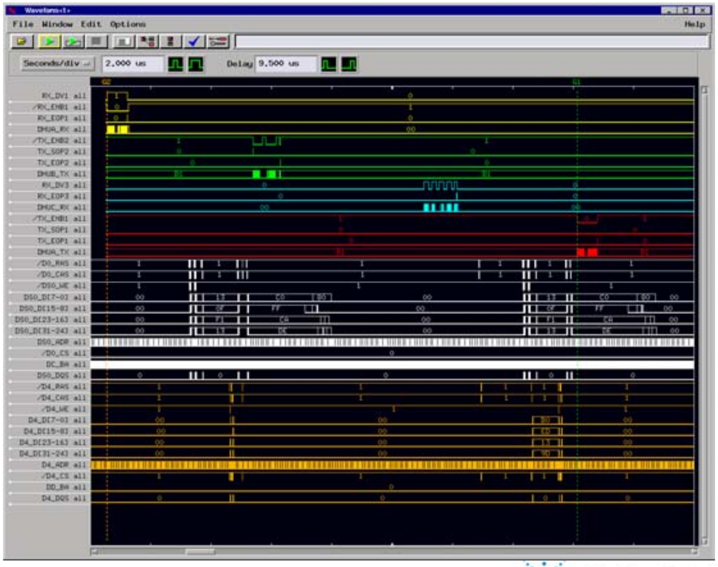
Cable loops port 2 TX into Port 3 RX, also channelized quad OC-3

Packet re-sent on OC-12 Port 1 TX

Agilent Technologies

This waveform display shows our packet propagating through the system. In white, the packet arrives on PORT A's receive interface. In cyan (light blue), the packet is sent on the PORT B transmit interface, then arriving on the PORT C receive, and finally sent on the PORT A transmit interface.

Correlation between FlexBus, DataStore, and Control Store



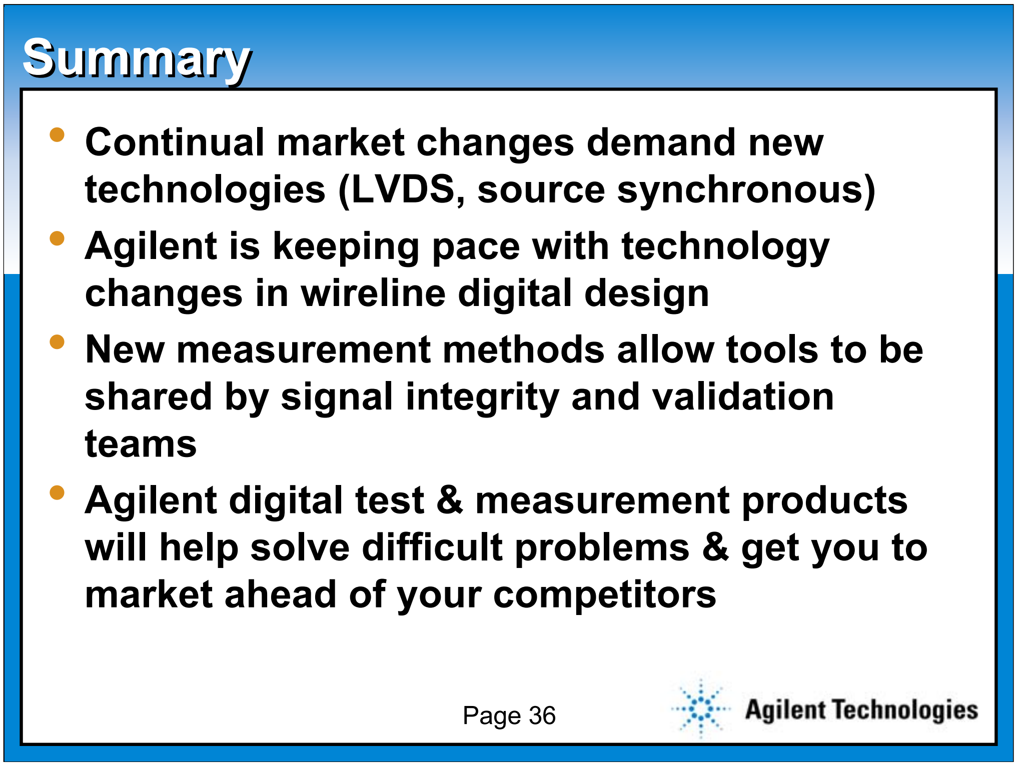
Agilent Technologies

This waveform shows our four packets on the FlexBus time correlated to the Data Store DDR memory (in white), and the control store DDR memory. You can see that the Data Store's "Write Enable" signal goes low after each packet arrives on the receive ports A and C (yellow and cyan). You can also see the read cycles occurring shortly before the packet transmissions on port B and A (green and red).

We now have enough information to track the flow of our packet in and out of data store memory and through the PHY chips, allowing us to pinpoint bottlenecks.

If at any point we want to see a decode of a packet we can use the Listing tool with additional decode software.

You'll see that not a lot of activity is happening on the control store in this example; no control information was changed during this measurement. However, if control plane behavior is of interest, you can see that the same principles of time correlation would allow you to debug those effects on your system.

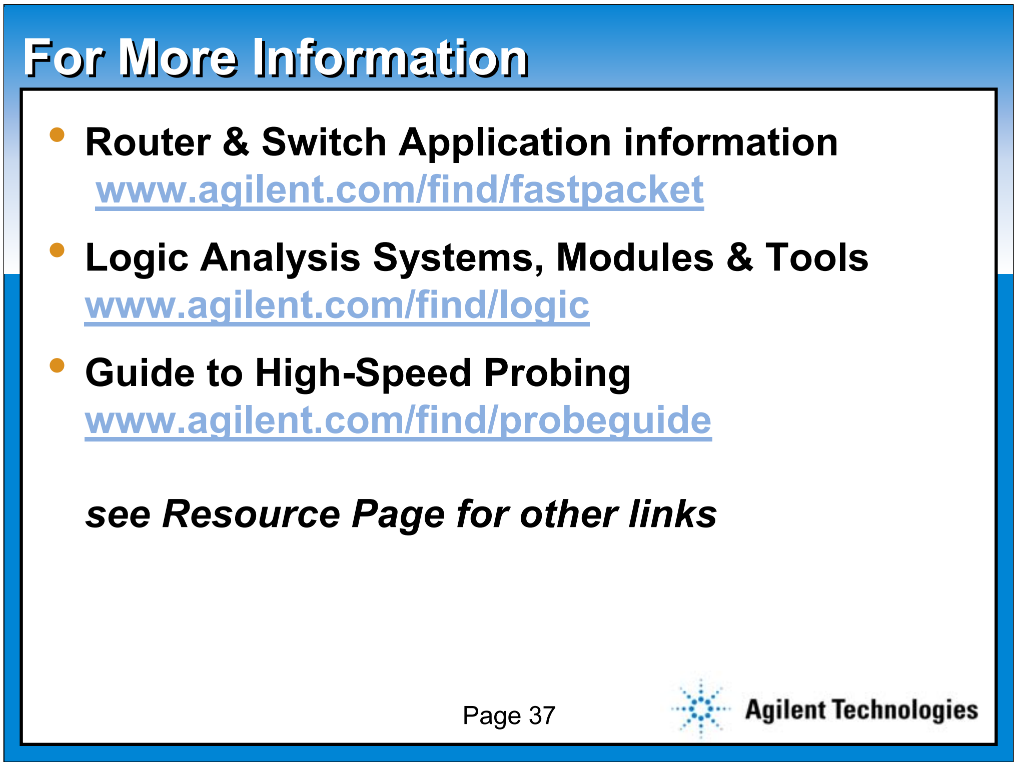


Summary

- Continual market changes demand new technologies (LVDS, source synchronous)
- Agilent is keeping pace with technology changes in wireline digital design
- New measurement methods allow tools to be shared by signal integrity and validation teams
- Agilent digital test & measurement products will help solve difficult problems & get you to market ahead of your competitors

Increased bandwidth requirements combined with heat dissipation, power and noise issues have brought forth new technologies such as LVDS and Source-Synchronous clocking. While these new technologies have rendered former test methods impossible, new methods have arisen to help. Agilent Technologies has kept pace by providing logic analysis products with not only sufficient acquisition speed, but with truly innovative measurement capabilities such as Eye Scan and Eye Finder.

Combining these new acquisition tools with Internet-focused bus analysis software and the system-wide perspective from time-correlated measurements will help you develop top-quality products and ship them ahead of your competitors.



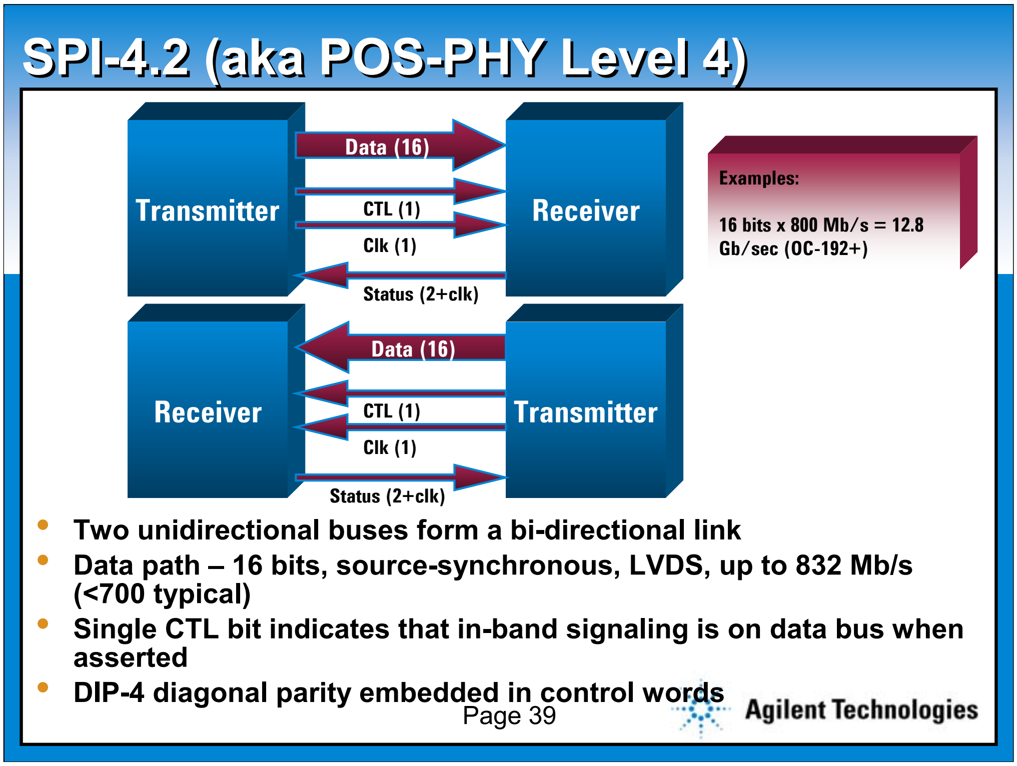
For More Information

- Router & Switch Application information
  www.agilent.com/find/fastpacket
- Logic Analysis Systems, Modules & Tools
  www.agilent.com/find/logic
- Guide to High-Speed Probing
  www.agilent.com/find/probeguide

see Resource Page for other links

Agilent Technologies

Backup Material



SPI-4.2 (aka POS-PHY Level 4)

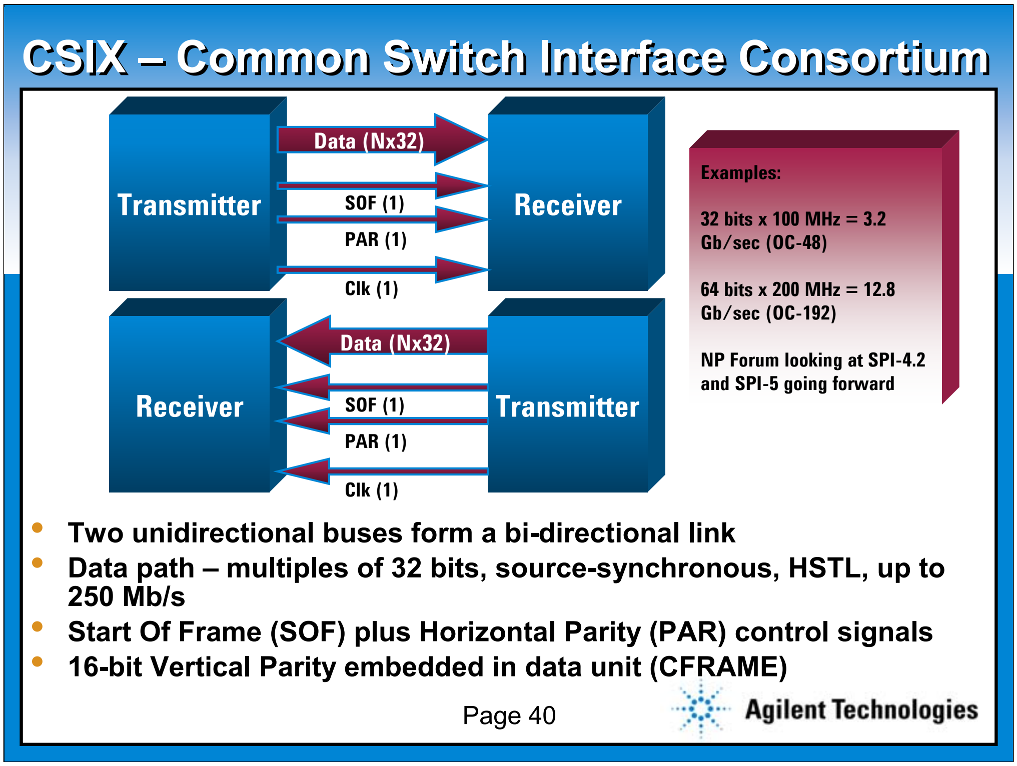
Examples:

16 bits x 800 Mb/s = 12.8 Gb/sec (OC-192+)

- Two unidirectional buses form a bi-directional link
- Data path – 16 bits, source-synchronous, LVDS, up to 832 Mb/s (<700 typical)
- Single CTL bit indicates that in-band signaling is on data bus when asserted
- DIP-4 diagonal parity embedded in control words

Agilent Technologies

This is a block diagram of the SPI-4.2 bus. There are really two unidirectional links. Each has a 16-bit data bus plus clock and framing/control bit, with LVDS signals. The clock is source-synchronous with speeds up to 832 Mbits/second. More commonly data rates between 622 Mbits/sec and 700 Mbits/sec are used, unless a lot of in-band control signalling is expected. In-band control means that all control information (such as start of packet, end of packet, parity, Idle) is sent on the data bus when the CTL signal is high. This significantly saves on pin and board routing resources, but does require some added bandwidth overhead to accommodate signaling without compromising the wire-speed transmission of packets.



CSIX – Common Switch Interface Consortium

Examples:

32 bits x 100 MHz = 3.2 Gb/sec (OC-48)

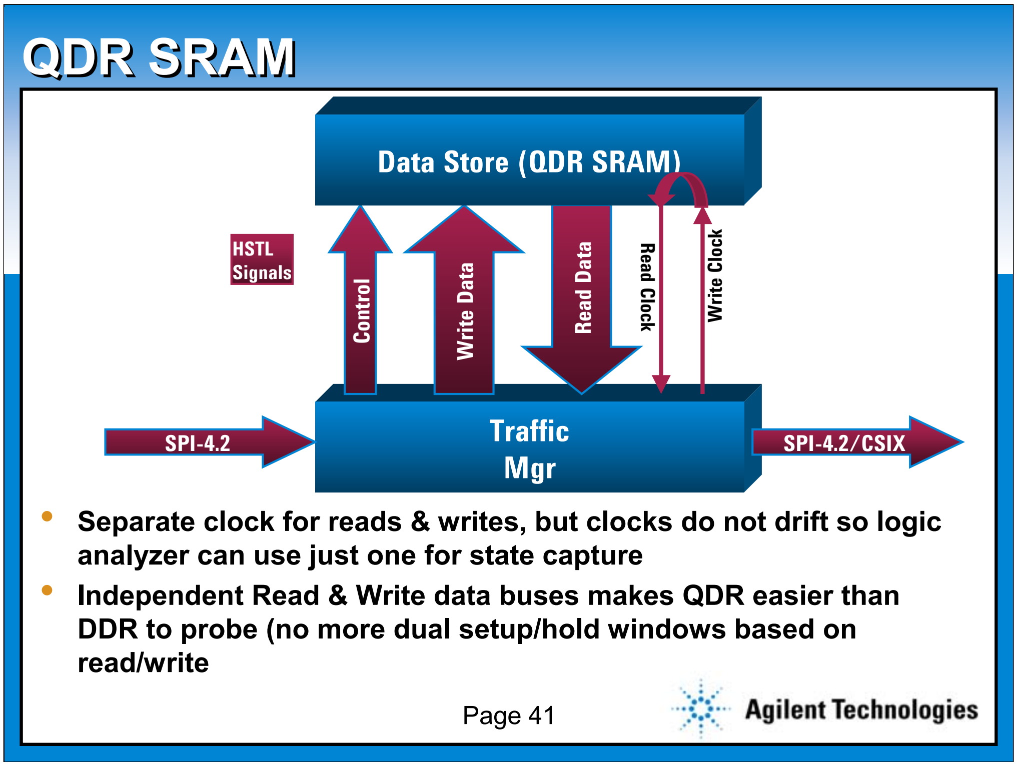
64 bits x 200 MHz = 12.8 Gb/sec (OC-192)

NP Forum looking at SPI-4.2 and SPI-5 going forward

- **Two unidirectional buses form a bi-directional link**
- **Data path – multiples of 32 bits, source-synchronous, HSTL, up to 250 Mb/s**
- **Start Of Frame (SOF) plus Horizontal Parity (PAR) control signals**
- **16-bit Vertical Parity embedded in data unit (CFRAME)**

The CSIX standard is much like UTOPIA and POS-PHY, except with source synchronous clocking and HSTL voltage levels. The bus width can be any multiple of 32 bits, depending on bandwidth needs. The clock rate is also flexible, between 100 and 250 MHz.

For OC-192 this spec requires a 64-bit data bus running at up to 200 MHz. The lack of scalability of this bus without real noise & routing troubles has led the Network Processing Forum to choose SPI-4.2 as their Switch Fabric interface bus for OC-192 speeds, but the CSIX CFRAME protocol will be used over SPI-4.2.



QDR SRAM

Data Store (QDR SRAM)

HSTL Signals

Control

Write Data

Read Data

Read Clock

Write Clock

SPI-4.2

Traffic Mgr

SPI-4.2/CSIX

- Separate clock for reads & writes, but clocks do not drift so logic analyzer can use just one for state capture
- Independent Read & Write data buses makes QDR easier than DDR to probe (no more dual setup/hold windows based on read/write

In many implementations of QDR SRAM, the separate Read and Write clocks actually come from the same source. The read clock is simply the write clock looped back to the controlling device. You can take advantage of this knowledge with Agilent's Eye Finder capability. Because the Read Data bus will have a skewed data valid window from the Write Data bus, Eye Finder can automatically discover the stable regions and you can use either the Read clock or the Write clock as your logic analyzer's state clock.

This is actually EASIER than probing DDR memory. DDR uses the same data bus for both reads and writes, but the setup & hold window is different depending on the type of bus cycle. While there is a DIMM socket probe available that handles these issues, many router designers use on-board DDR's, not in DIMM sockets, and a DIMM probe is inconvenient.