

Debugging and Validating PCI Express I/O

With these tips and tricks for using a logic analyzer, you can speed time to market and increase confidence in your design.

by Richard Markley
Logic Analysis Product Planning Manager
Agilent Technologies
richard_markley@agilent.com

Marco Davila
R&D Hardware Designer
Agilent Technologies
marco_davila@agilent.com

As PCI Express continues to replace PCI in many designs, engineers are finding themselves in uncharted territory. High-speed serial links running at 2.5 Gbps introduce new challenges that were not seen with traditional wider and slower parallel buses like PCI. Vias look like stubs. Data is 8b/10b encoded such that clocks are embedded. Signal swings are minimal. The list goes on and on. With these new challenges, you will need to rely more on test equipment than you have in the past.

One of these key pieces of test equipment is the logic analyzer. Although at first glance a logic analyzer may not appear to be suited for debugging a serial bus, recent advances have made the logic analyzer a powerful tool for system bring up and validation of serial buses like PCI Express (PCIe).

New technologies allow the logic analyzer interface (also known as an analysis probe) to use its hardware resources (instead of the logic analyzer's triggering resources) to look for packets.

Probing Advancements

Successfully probing a PCIe link is not a trivial task. Because of the gigabit speeds, test and measurement vendors need probing that is non-intrusive and easy to use.

The simplest method to probe a PCIe link is to use a slot interposer. Slot interposers require no forethought when it comes to probing – you simply plug the interposer into an available PCIe slot and plug your add-in card on top. Although they are simple to use, some interposers

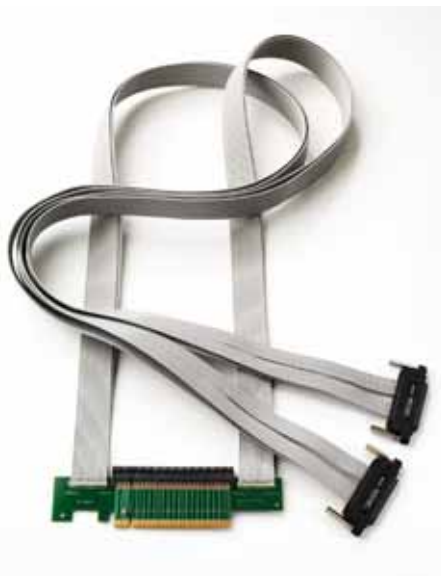


Figure 1 – PCI Express slot interposer

are less intrusive than others. Obviously, an interposer cannot be so electrically intrusive that it breaks the link (that is, it doesn't allow the device under test to work). However, it is also important to pay attention to the mechanical intrusiveness of a slot interposer. Interposers that are shorter, with vertical egress (see Figure 1), provide more testing options to system designers.

Although interposers are simple to use, they are not helpful for chip-to-chip designs. Probing these designs (often called “midbus probing”) typically requires a designed in footprint. The PCI-SIG has

specified a common footprint for all test vendors. This footprint is a “connector-less” design that uses landing pads for probing. Although very different from a slot interposer, the same potential concerns exist – electrical and mechanical non-intrusiveness.

In addition to these potential concerns, many designers should also consider how easy the probes are to use. Do they require special cleaning to get a reliable connection? Are they compatible with multiple board finishes such as hot air solder leveling



Figure 2 – PCI Express midbus probe

process (HASL) or gold plating? Do they require external cooling fans? An example of a midbus probe is shown in Figure 2.

Although a midbus probe is typically the preferred method for probing chip-to-chip designs, it does require a footprint to be designed in. Sometimes engineers do not have the room for a design in footprint, or they may have not considered debugging and validation early enough to design in the footprint. In these cases, a flying lead set can be very beneficial. As with all probing systems, the flying lead set must be electrically and mechanically non-intrusive. It should allow designers to

probe at the full link speed (2.5 Gbps) while keeping probe head volume to a minimum. An example of a flying lead set is shown in Figure 3.

Triggering Advancements

Because of the parallel nature of the logic analyzer, triggering on a packetized bus requires you to use many of the logic analyzer's triggering resources to define just the start of a packet. This is especially true in PCI Express, which has the option of multi-



Figure 3 – PCI Express flying lead set

ple lane widths. The serial nature of the bus makes triggering significantly different from triggering on a parallel bus, where you would normally specify a value for a specific label.

New technologies allow the logic analyzer interface (also known as an analysis probe) to use its hardware resources (instead of the logic analyzer's triggering resources) to look for packets. These packet analysis probes contain “packet recognizers” specifically designed to help trigger on serial links. These allow you to define as many as four packets in each direction for the logic analyzer to trigger on. In addition, each packet recognizer allows you to define the entire

packet header, and as many as 8 bytes of the data payload (for a 3 double word [3DW]). These packet recognizers also provide the means for specifying “don’t cares” within the header/data fields. This stands in stark contrast to traditional logic analyzer resources that only allow you to define the packet type (transaction layer packet [TLP] or data link layer packet [DLLP]).

At first, the packet recognizer must determine the start of the packet. The packet may start in one of four lanes for a x16 link (lane 0, 4, 8, or 12), so the packet recognizer must look in each of these lanes. It does this automatically – you do not have to worry about defining the trigger steps to recognize this. Traditional logic analyzer triggering ends up using a large portion of its resources to determine only this event.

After resolving the start of packet and deskewing the lanes (just as the actual receiver does), the packet recognizers then look for matches to fields within the packet header and the data payload. The packet analysis probe will then send a signal back to the logic analyzer, which it can use in a trigger. These signals can be used with the full triggering resources of the analyzer (including counters, timers, sequencers, storing, and multi-way branching) to provide very robust, powerful triggering.

Common Debug Triggers

Using packet recognizers allows you to define an almost limitless amount of triggers. They are often used in debug techniques such as:

- Prestore and qualified capturing of packets
- Cross-bus triggering
- Triggering using an exerciser

During initial bring up of a PCIe device, you may want to capture a specific event and a large period of time before that event. Because you need to capture a long period in time, it is often beneficial to only store events that are of interest in the logic analyzer’s memory. However, this requires additional triggering and storage resources. If these resources are completely used in defining the type of packet, this may not be possible.

...test equipment like logic analyzers can help you as you move from the parallel world to the serial world.

A packet recognizer helps alleviate this problem. For example, you can define a specific packet header along with several bytes of data. We will call this “3DW with Data.” You can then define another packet that includes all of the types of events you want to store. In this case we only want to store other TLPs – all other fields in the recognizer are left as “don’t cares.” We call this “TLP only.” The logic analyzer will then use a simple pattern trigger to find the “3DW with Data” event, and you now have all of the analyzer’s resources left to qualify what is stored.

Often you will only want to see information before the trigger. In this case, you can set the logic analyzer to do what is called “prestore.” A 100% prestore will only store information before the trigger, so you can capture a larger period of time before your trigger event. When used in conjunction with the default storing, this allows you to capture the maximum amount of time before the trigger. In most logic analyzers, you can easily define the percent of “pre” or “post” store.

In a serial architecture like PCI Express, a disagreement between the perceived traffic viewed by the transmitter and receiver doesn’t always point to the root cause of a problem. Using a cross-bus triggering technique allows you to not only trigger on this disagreement, but also locate the source of problem. This problem might be caused by another bus in the system such as the processor system bus, DDR memory bus, SATA/SAS bus, or another I/O bus.

This is a very easy trigger to setup, but very powerful in the information that it provides. You can trigger from any one bus and capture time-correlated events on the other buses in their system. For example, a common trigger involves looking for a bus hang on the processor system bus.

This will then trigger and capture data on all of the additional buses you are looking at. Should the processor bus hang be caused by an event on the PCIe link, this is a quick way to see the events time-correlated together for maximum debug.

Another common cross-bus triggering technique involves looking at the PCIe link from the south bridge to a switch with multiple PCI slots. For example, it is often beneficial to trace a specific event as it occurs on the PCI bus and travels through the bridge to the PCIe link. Once again, packet recognizers can be very beneficial in this case, because they allow you to look for a very specific packet header with data. Traditional triggering using the logic analyzer’s resources would have a difficult time defining the packet with enough detail to capture this event easily.

Another common debug technique involves using an exerciser to generate traffic on the PCIe link while using the logic analyzer to capture the response to this stimulus. This is often known as “stimulus and response capture” and is a very powerful technique that is normally employed later in a designer’s program to test the compliance of their devices.

Conclusion

PCI Express is taking off as a common I/O interconnect for many designers. Although it has many benefits (scalable, backwards compatibility to PCI, fewer signals), it does present some significant design challenges. Because of this, test equipment like logic analyzers can help you as you move from the parallel world to the serial world.

To learn more about the equipment discussed in this article, please visit www.agilent.com/find/pciexpress or contact your local Agilent field engineer. 