



Strategies for Debugging Serial Bus Systems with Infiniium Oscilloscopes

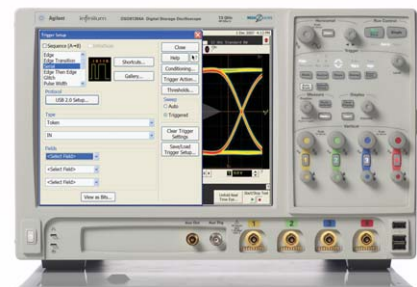
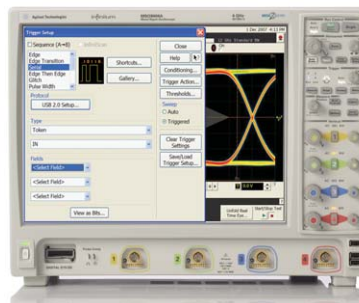
Application note 1611

Introduction

Agilent Infiniium Oscilloscopes provide many of the features of protocol analyzers such as decode listings and protocol specific triggering. However, a good question is “If protocol analyzers already exist, why is this capability needed?”

First of all, having a single instrument that provides the capabilities of an oscilloscope, logic analyzer and protocol analyzer provides lots of advantages. First of all, there is the convenience of bench space. Why crowd your bench with three instruments when one will do? Secondly, even if you happen to have all three instruments handy, it is a hassle to turn on another instrument and set it up. A single “All-in-one” instrument saves time.

Secondly, having a single instrument that provides both serial and analog capabilities means that serial problems can be traced back to their root cause. For example, if there is inter-symbol interference on a PCI-Express bus, the problem will show up as an invalid packet (and invalid symbol) in the decode listing. Just point at the problem in the listing and the corresponding waveform is shown on the screen. Without both serial and analog capabilities, it is difficult to tell what caused the invalid symbol.



Another benefit of using oscilloscopes to debug serial buses is easy connectivity. There’s no need for a standard port or special IO for connectivity. In addition, oscilloscope probes are passive while protocol analyzers typically provide re-transmit and re-timing. If physical layer problems exist, connecting the protocol analyzer can mask them.

Finally, Infiniium Oscilloscopes are able to support a wide variety of protocols such as I²C, SPI, RS232, USB and PCI-Express. So the oscilloscope doesn’t just replace a single protocol analyzer; it replaces a number of them. It is very common for Engineers to attempt to “mentally decode” waveforms because they don’t have the right protocol analyzer at hand. This new capability eliminates this difficult process forever.

Who Should Read This Application Note?

This application note is for digital designers in R&D working with both analog and digital components, including microcontroller and DSP systems using serial buses. This application note discusses the challenges associated with and new solutions for debugging serial bus designs including PCI-Express Generation 1, Inter Integrated Circuit (I²C), Serial Peripheral Interface (SPI), or Universal Serial Bus (USB).



Agilent Technologies

Key Functionality

In order to debug a serial bus, it is essential to have both decode as well as triggering. Depending upon the protocol, the meaning of “decode” can mean a variety of things. For example, in a high speed serial protocol like PCI-Express, decode can mean anything from symbols embedded in the waveform to a packet listing. In lower speed protocols like SPI, decode is simply a listing of the payload values.

Embedded Decode

Embedded Decode utilizes the oscilloscope’s waveform display to overlay the decode information in the graticule. This provides very tight correlation between a waveform and the corresponding decode. This is functionality that protocol analyzers do not provide. It also eliminates the common problem of having to “mentally decode” the waveform. (It may sound strange, but many Engineers have told me that they do this on an oscilloscope). An example of an embedded decode from an Agilent Infiniium 9000 series oscilloscope is shown in Figure 1.

Note that for high speed protocols, the embedded decode can either be in symbols (such as 8B/10B symbols) or packets. This is two different levels of abstraction. An example of embedded symbols is shown in Figure 2.

Decode listing

A listing is a very efficient method for displaying data because it shows a complete packet on a single line as shown in Figure 3. This is the display that is used on protocol analyzers. However, on an oscilloscope, there is correlation between the waveforms and the selected packet. Notice in Figure 3 there is a selected packet (highlighted in blue) and there is also a blue line in the waveform display showing the corresponding waveform.

Some information can easily be shown in a column of a table but some cannot. The most obvious example is a payload, which can consist only of a couple bytes or dozens of bytes. In order to be able to view a payload, use the Payload Tab as shown in Figure 4. This shows all of the bytes in this payload (which happens to be high speed USB) along with an ASCII display. Having a separate Payload Tab is essential because it allows entire payloads to be displayed regardless of length. Also, users can change the base (hex, binary, or decimal) to whatever format is preferred for them.

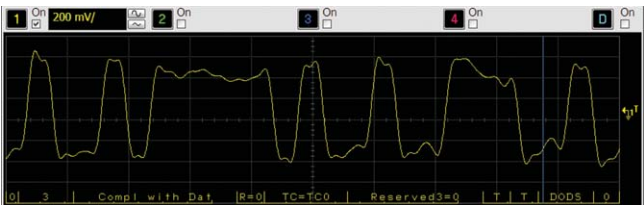


Figure 1. An example of embedded decode of PCI-Express fields

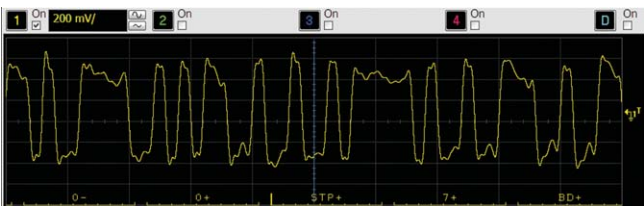


Figure 2. Example of embedded decode of 8B/10B symbols

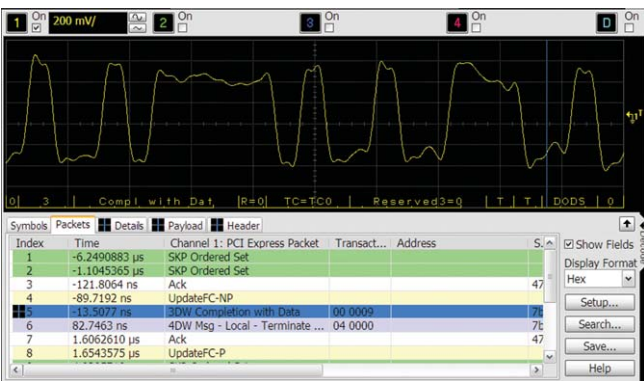


Figure 3. A decode listing of packets correlated with waveforms



Figure 4. A payload tab shows far more information than can fit in a column of a table

Key Functionality (continued)

Decode listing

When we did research on how Engineers like to see serial data, we found that they could not agree on a single format to be used all of the time. I showed the Engineers a variety of formats and asked them which one they preferred. Instead of choosing one, they requested all of them. The point they made is that a single view of the data does not cover all situations and they wanted a variety of formats to give them as much flexibility as possible. Some relatively simple protocols like I²C, RS232 or SPI, require nothing more than a listing of Packets and a Payload Tab. But high speed protocols like PCI-Express, SATA, SAS, and MIPI require more views of the data.

One of the formats that users requested is the Header Tab shown in Figure 5. This shows the header broken down by fields in format similar to the one used in protocol specifications. Having a view of the data that matches the way the Engineers were taught the protocol is very important.

Another format that the Engineers requested is a hierarchical view called the Details Tab. This recognizes that there can be a hierarchical in protocols (such as fields at different layers of the protocol). An example is shown in Figure 6 where the Physical, Data Link and Transaction Layers can clearly be seen.

The last view that engineers requested is a symbol view. An example is shown in Figure 7.

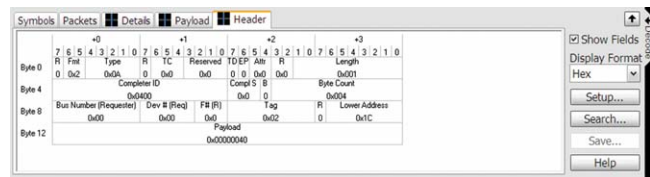


Figure 5. A heading view of a packet that matches the way packets are shown in the protocol specification



Figure 6. A Details Tab clearly shows the Physical, Data Link and Transaction Layers of a high speed protocol

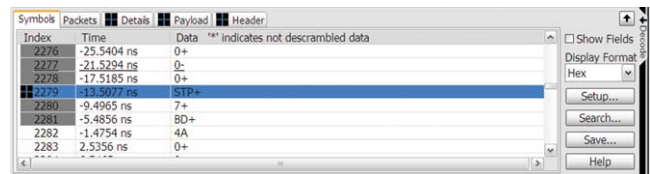


Figure 7. An example of a Symbol Listing

Key Functionality (continued)

Triggering

While doing research on oscilloscopes and serial protocols, we asked Engineers if serial decode or triggering was more important. Most Engineers simply could not answer the question because both are essential to debugging a serial bus. After all, what good is it to see the bus if you can't find what you want? That is why oscilloscopes provide both serial decode and triggering.

Every oscilloscope user is familiar with common triggers such as edges and pulse widths, but an entirely new suite of triggers is required for serial. It is not enough to simply trigger on some aspect of the waveform, but the oscilloscope needs to trigger on aspects of the protocol itself. For example, a common trigger that users desire for USB would be on IN tokens. Or, for I²C, an example would be a Read that occurs as result of a restart. These are known as packet types because they allow users to trigger only on a specific type of packet. An example is shown in Figure 8.

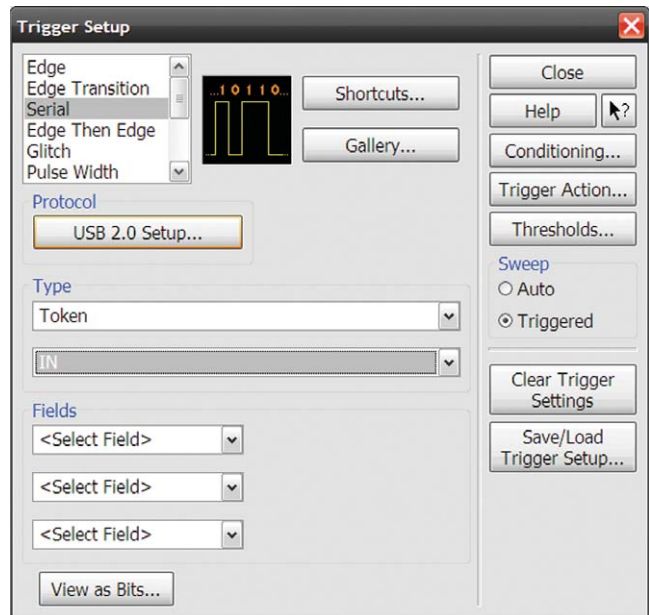


Figure 8. An USB trigger on all packets where the Token is IN

However, there are many cases where simply triggering on a packet type simply isn't enough. It is essential to be able to trigger on specific fields within a packet just like protocol analyzers do. An example is to trigger on a specific data and address value in I²C as shown in Figure 9.

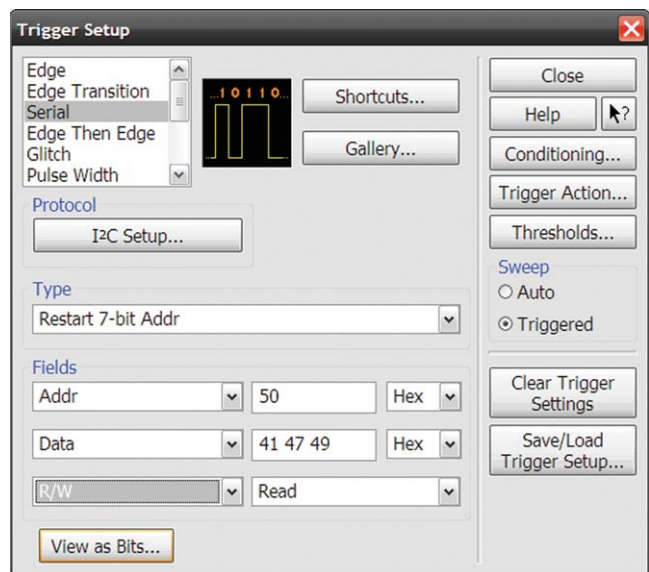


Figure 9. An I²C trigger on a Read of Address 50 with the Data 41 47 49

Key Functionality (continued)

While we were doing our research, we discovered that even though Engineers wanted the ability to specify packets and fields as triggers, they also wanted the ability to see exactly what the definition of the trigger is. That is the purpose of the “View as Bits” dialog shown in Figure 10

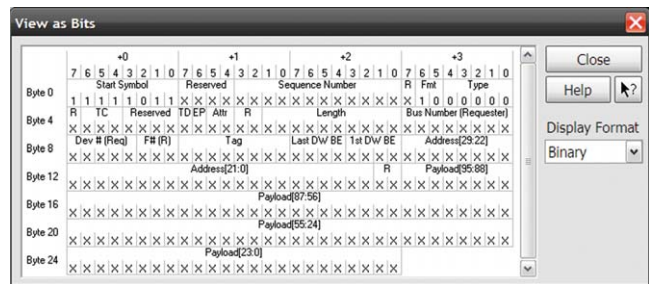


Figure 10. View as Bits dialog shows the “bit by bit” definition of the serial trigger

A final type of triggering is referred to as the Symbol Sequence. Once again, this is for high speed serial buses. This is for the case where the Engineers want to enter a list of specific symbols to trigger on. An example is shown in Figure 11. It begins with a K code (SDP), ends with a K code (END) but uses data values for the rest of the sequence.

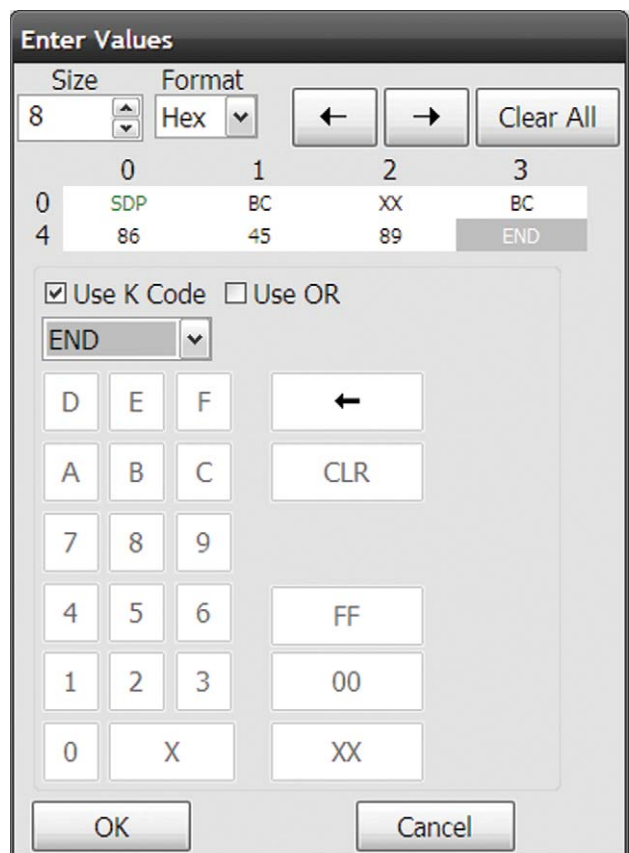


Figure 11. An example of a Symbol Sequence

Setting Up a Serial Bus

Setting up a serial bus on an oscilloscope is not necessarily simple. There are a variety of factors that must be specified correctly in order for both triggering and serial decode to work. These factors include sample rate, memory depth, trigger levels, measurement thresholds and clock recovery (high speed buses only). For simplicity, the Infiniium 9000 series oscilloscopes include an Autoset button that will automatically setup all of these parameters for the user.

Sample Rate

In general there is an optimal sample rate defined by the speed of each serial bus. Naturally, if the sample rate goes below this level, the serial decode cannot be performed. Traditionally, however, oscilloscopes have optimized their sample rate based upon the current setting of the time per division. This optimizes the view of the current waveform (which is best for finding signal integrity problems). This is a problem when viewing serial decode because zooming in on a waveform will cause the serial decode to disappear. Therefore, until users want to investigate a signal integrity issue, it is recommended that the sample rate be set to a fixed point.

Memory Depth

Memory Depth is another key issue. If the depth is very large, this can slow down the update rate of the oscilloscope. However, making the depth too small can result in the trace consisting of a single fragment of a packet (which cannot be decoded). Ironically, the serial triggering hardware will still work fine because it sees the complete packet even though the depth is not great enough to save it. In many cases, the users will need to experiment to find their optimal memory depth. While triggering helps them to find a packet of interest, it is also essential to have a fair amount of context around the packet to see what happened before and after.

Trigger Levels

Trigger Levels are familiar to all oscilloscope users because they are used in virtually all oscilloscope triggers. Serial triggering is relatively simple because if the trigger is set to the middle of the waveform it will generally work fine. Trigger Levels are probably the easiest issue to deal with in serial debugging.

Measurement Thresholds

Measurement Thresholds are very similar to Trigger Levels even if they are somewhat less familiar. The biggest confusion with Measurement Thresholds is how they are used. Trigger Levels are used by the oscilloscope hardware to trigger. Measurement Thresholds are used by the oscilloscope software to figure out the decoding. The two functions are independent; i.e. an oscilloscope that is triggering just fine on a serial bus may fail to decode. Compounding the problem is that Trigger Levels are prominently displayed in the main window of every oscilloscope, while Measurement Thresholds are buried. To alleviate this problem, the Infiniium 9000 series automatically sets up both the Trigger Levels and Measurement Thresholds whenever the Autoset button is pressed. However, if the user manually changes a Trigger Level (such as by turning the front panel knob), serial triggering may no longer work. In a similar manner, if a user changes any of their Measurement Thresholds, serial decode may no longer work. The best advice for most Engineers is if either triggering or decode doesn't work, press the Autoset button again. In most cases, that will take care of the problem.

Clock Recovery

Most High speed serial buses use clock recovery instead of a separate clock signal. The Autoset function will attempt to choose a method for a specific protocol, but it may be up to the user to specify the correct method.

Limitations

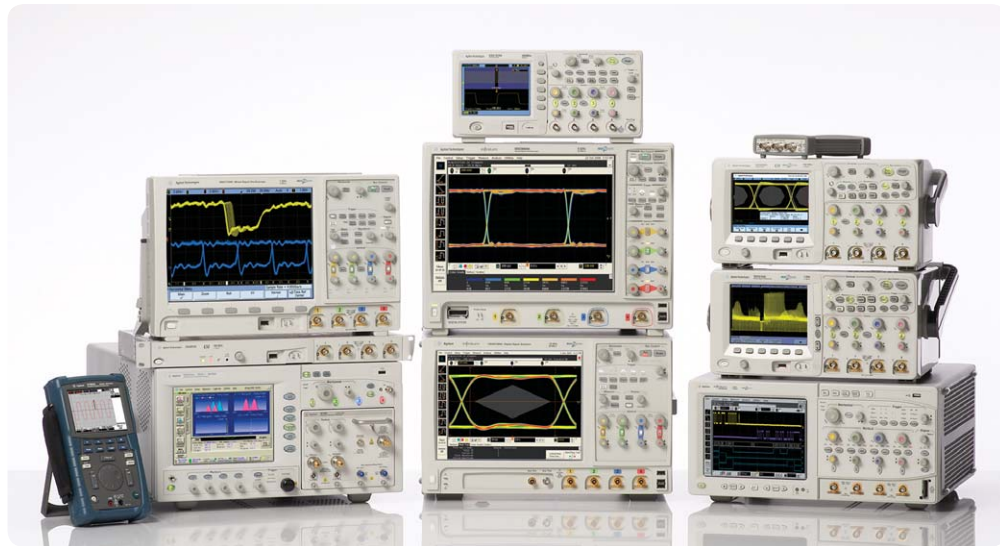
While Infiniium Oscilloscopes can perform many of the functions of a protocol analyzer, there are some limitations. The biggest is memory depth. Oscilloscopes acquire samples to be able to recreate waveforms while protocol analyzers only need to recreate packets. Therefore, the number of packets that can be stored by protocol analyzers is vastly greater.

A second limitation of its performance. Because protocol analyzers have dedicated hardware to decode packets while oscilloscopes use software decode, for large memory depths the oscilloscopes will be sluggish. (Note that this limitation does not apply to the 5000 and 7000 series oscilloscopes which have dedicated hardware for serial decode).

Protocol Analyzers often have specialized performance analysis capabilities that go well beyond just triggering and decoding. For these capabilities, a protocol analyzer is required.

Summary

New advances in Infiniium Oscilloscopes have enabled Engineers to debug serial buses as well as analog and digital signals. In essence, they have become a combination of an oscilloscope, logic analyzer and a protocol analyzer all in the same instrument. For example, the Agilent Infiniium 9000 series oscilloscopes have incorporated many of the most important protocol analyzer features such as a decode listing and protocol specific triggering. This means Engineers no longer have to clutter their bench with a wide variety of instruments or waste time trying to correlate results from one instrument to another. These new serial capabilities mean the most common serial debugging can be achieved on the oscilloscope for a wide variety of protocols such as I²C, SPI, RS232, USB and PCI-Express.



Agilent Technologies Oscilloscopes

Multiple form factors from 20 MHz to >90 GHz | Industry leading specs | Powerful applications



Agilent Email Updates

www.agilent.com/find/emailupdates

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



Agilent Direct

www.agilent.com/find/agilentdirect

Quickly choose and use your test equipment solutions with confidence.



www.agilent.com/find/open

Agilent Open simplifies the process of connecting and programming test systems to help engineers design, validate and manufacture electronic products. Agilent offers open connectivity for a broad range of system-ready instruments, open industry software, PC-standard I/O and global support, which are combined to more easily integrate test system development.



www.lxistandard.org

LXI is the LAN-based successor to GPIB, providing faster, more efficient connectivity. Agilent is a founding member of the LXI consortium.

Windows® is a U.S. registered trademark of Microsoft Corporation.

Remove all doubt

Our repair and calibration services will get your equipment back to you, performing like new, when promised. You will get full value out of your Agilent equipment throughout its lifetime. Your equipment will be serviced by Agilent-trained technicians using the latest factory calibration procedures, automated repair diagnostics and genuine parts. You will always have the utmost confidence in your measurements.

Agilent offers a wide range of additional expert test and measurement services for your equipment, including initial start-up assistance onsite education and training, as well as design, system integration, and project management.

For more information on repair and calibration services, go to

www.agilent.com/find/removealldoubt

www.agilent.com

www.agilent.com/find/xxxx

For more information on Agilent Technologies' products, applications or services, please contact your local Agilent office. The complete list is available at:

www.agilent.com/find/contactus

Americas

Canada	(877) 894-4414
Latin America	305 269 7500
United States	(800) 829-4444

Asia Pacific

Australia	1 800 629 485
China	800 810 0189
Hong Kong	800 938 693
India	1 800 112 929
Japan	0120 (421) 345
Korea	080 769 0800
Malaysia	1 800 888 848
Singapore	1 800 375 8100
Taiwan	0800 047 866
Thailand	1 800 226 008

Europe & Middle East

Austria	01 36027 71571
Belgium	32 (0) 2 404 93 40
Denmark	45 70 13 15 15
Finland	358 (0) 10 855 2100
France	0825 010 700
Germany	07031 464 6333
Ireland	1890 924 204
Israel	972-3-9288-504/544
Italy	39 02 92 60 8484
Netherlands	31 (0) 20 547 2111
Spain	34 (91) 631 3300
Sweden	0200-88 22 55
Switzerland	0800 80 53 53
United Kingdom	44 (0) 118 9276201
Other European Countries:	

www.agilent.com/find/contactus

Revised: October 1, 2008

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

© Agilent Technologies, Inc. 2009
Printed in USA, June 1, 2009
5990-4093EN



Agilent Technologies