

Case Study:

Solving An Intermittent Serial Communication Issue Using Advanced Triggering and Filtering



Case Study

The Scenario

This design engineer for a large mil-aero company in San Diego is testing an early release of an avionics system. He needs to pass the sign-off on a major milestone for the project. But, his customer (DOD) has been sitting in on the tests and see's something that does not seem right. The customer has to sign off before the design can be put into production. The design engineers first reaction is to pull out his oscilloscope and troubleshoot the system.

In order to detect this problem, our design engineer needs to monitor many of these IO points (Fibre Channel in this case) simultaneously. His scope and logic analyzer don't "speak" Fibre Channel and his logic analyzer can't interface with the IO. Even worse, neither the scope nor the logic analyzer have advanced triggering. He wonders if he has a firmware bug, or could this be caused by a hardware problem? He decides that the best course of action is to use his Absolute Analysis Investigator™ and oscilloscope together so that he can get a better view of what is happening. First, he tries using his oscilloscope to trigger on the problem and cross triggers the Investigator™ to see which code is being transmitted at that time.

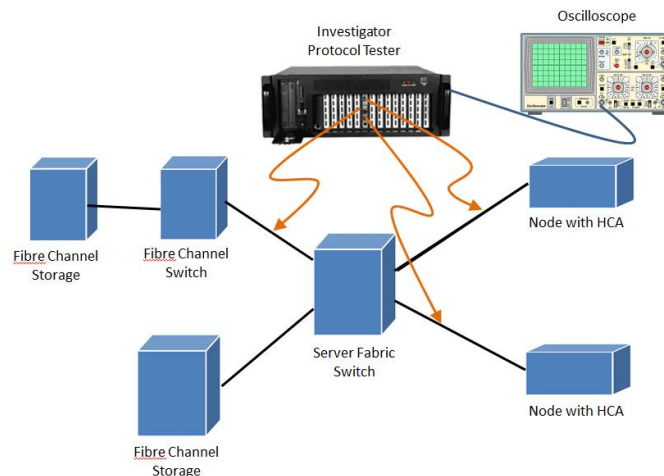


Figure 1 - Investigator Monitoring Multiple Fibre Channel Links Simultaneously

Isolating the Problem Using Advanced Triggering

Our design engineer finds that three different sets of commands are being transmitted about the same time as the problem. Even worse, there's not a 100% correlation between any command and the problem. What is causing this event? Why doesn't it happen consistently? So, he turns the situation around and uses the Investigator to trigger on each command individually then cross trigger the scope. Using this technique, he finds that when one of the commands is sent, the problem occurs about 20% of the time. The other commands don't seem correlated.

It appears that this problem is rate sensitive. So, he sets the Investigator to trigger only when the command is sent at a rate greater than

3times/sec. He now finds that this gives him a 100% correlation with the problem. Below 2

times/sec the problem doesn't occur at all. Ok - the problem is definitely rate sensitive. He then uses the Investigator to trigger the oscilloscope and take data only when the command occurs at a rate greater than 3x/sec. By doing this, he is able to quickly isolate the problem to a circuit that was not able to reset fast enough before the command was repeated.

Conclusion

It took our engineer a few days to adjust the design and get his customer to understand the solution to their very justified concern and to sign off. But, his company did end up completing the project many man days were saved and confidence in design restored.

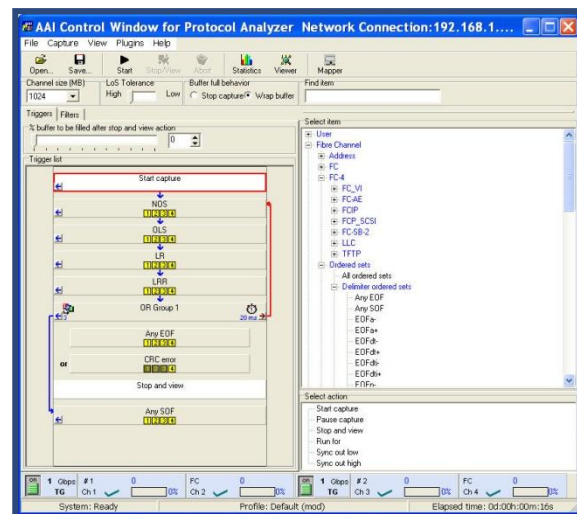


Figure 2 - Investigator Trigger Screen Example

Next Case Study

In our next *Digital Engineering and Debug* case study, we'll feature interoperability debug at the primitive level. We will send it to your email address on record once it is ready.

If you have an interesting case study you would like to contribute to our serial test community at large, please email us using the information below.

Also, for more information on Investigator, or to [schedule a demo](#) on how our serial validation test equipment can help your design team, please contact us:

Absolute Analysis

Phone: (805)376-6048

Web: www.AbsoluteAnalysis.com