DEBUGGING SPACEWIRE DEVICES USING THE CONFORMANCE TESTER

Session: Test & Verification

Short Paper

Dr Steve Parkes, Dr Martin Dunstan

School of Computing, University of Dundee, DD1 4HN, Scotland, UK

E-mail: sparkes@computing.dundee.ac.uk, mdunstan@computing.dundee.ac.uk

1 INTRODUCTION

The SpaceWire Conformance Tester is a device developed by STAR-Dundee for ESA to perform black-box testing of SpaceWire devices against the SpaceWire ECSS-E-50-12A standard [1]. The tests performed by the Conformance Tester are those which can be performed over a single SpaceWire link to the unit under test without cooperation from that unit. Thus link initialization behaviour and the response to data and control characters can be investigated while PCB layout and connector compliance cannot. The tests to be performed are launched from easy-to-use software running on the host PC with concise test results providing important feedback on how the unit under test performed.

This paper examines two of the errors/issues that have been identified with different SpaceWire devices through the use of the SpaceWire Conformance Tester.

This paper also introduces a novel test procedure which measures the speed at which the unit under test recovers from link errors at different points of the link initialisation process. The graphical output of this test can be used to identify anomalous behaviour of a SpaceWire device which might be hard to detect in other ways and to provide measurements of the link initialisation timing parameters.

2 MEASURING LINK START-UP SPEED

Clause 6.6.5 of the SpaceWire ECSS-E-50-12A standard specifies that after a reset or disconnect the SpaceWire link transmitter shall initially commence operating at a data signalling rate of (10±1) Mb/s and shall operate at this rate until commanded to operate a different data signalling rate. Clause 6.6.6 specifies that the operating rate shall not be changed before the link has reached the *Run* state.

To validate the (10±1) Mb/s link initialisation speed the test link interface (TLI) of the Conformance Tester is instructed to restart the link to the UUT but is forbidden to transmit any FCTs. A compliant UUT will begin error recovery and link initialisation eventually reaching the *Connecting* state. Since the TLI is forbidden to send FCTs, the UUT must return to *ErrorReset* after the nominal 12.8 µs time out period. From the time at which the UUT enters the *Started* state and begins transmitting NULLs to the time at which it stops the link to move to *ErrorReset* the UUT must operate at a data signalling rate of (10±1) Mb/s. It ought to be a trivial matter to measure the link rate of the UUT and confirm that it lies within the specified limits.

However, the standard doesn't specify the conditions under which the data rate limits are to be satisfied. Should the duration of every bit transmitted during start-up lie within 90.9 ns to 111.1 ns? Should the duration of every recovered transmit clock period lie within 181.8 ns to 222.2 ns? Should the signalling rate be averaged over multiple bits: how many bits? When does link initialisation end and shutdown begin?

The Conformance Tester software will report the minimum and maximum bit-to-bit rates, the minimum and maximum clock period from rising-edge to rising edge and from falling-edge to falling-edge, and the bit rate averaged over all bits received. If any bit-to-bit rate lies outside the (10±1) Mb/s range the UUT is considered to have failed. It is the responsibility of the user to examine the reported limits to determine whether the test failure corresponds to a violation of the standard. By applying a stringent interpretation of the standard we encourage the user to investigate the issue.

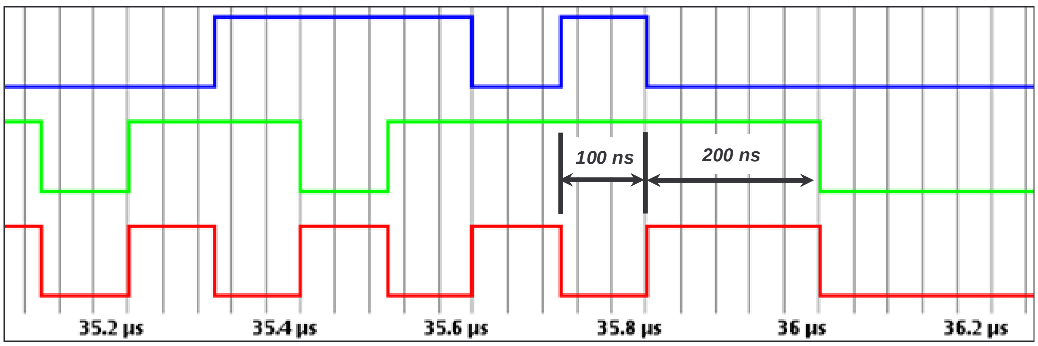

Figure 1: link shutdown with the strobe signal held for an extra bit period

An example of a test failure which has been observed with several UUTs is shown in Figure 1. The UUT has held the strobe signal stable for an extra bit period while stopping the data signal during link shutdown to avoid a simultaneous D/S transition. This doubles the duration of the last bit which causes a violation of the bit-to-bit rate. Since the failure applies to the last bit of the transmission during link shutdown one could argue that the (10±1) Mb/s rate does not apply. However, by reporting a test failure the user is made aware of an issue that not be detected otherwise.

3 EMPTY PACKET CREDIT COUNTING FAILURES

Clause 8.3 of the standard specifies that the transmitter shall not transmit any N-Chars unless its credit counter is greater than zero. The transmit credit counter shall be decremented by one for each N-Char transmitted and shall be incremented by eight for each FCT received. A similar counter is maintained by the receiver. Clause 8.2.1 states that data characters as well as EOP and EEP are N-Chars. Clause 8.9.3.2 states that empty packets may be silently discarded.

As a result of these clauses being considered in isolation, the Conformance Tester has discovered UUTs which discarded empty packets before updating their receive credit counter. If a UUT suffers from this problem then empty packets received by the UUT will reduce the effective size of the UUT receive buffer. By sending a non-empty test packet to the UUT followed by a number of empty packets the Conformance Tester is able to detect this bug. If the bug is present the Conformance Tester can also report the size of the UUT receive buffer.

4 NULL ARRIVAL TIME TESTING

The NULL arrival time test is a recent addition to the Conformance Tester suite and is one of the most powerful for probing the UUT.

Consider the following: the Conformance Tester TLI is connected to a UUT with the link in the *Run* state. At time T_0 the TLI sends a parity error to the UUT. A compliant UUT will respond by moving to *ErrorReset* and begin the link initialisation process. Let the TLI ignore the UUT disconnection and remain in the *Run* state transmitting NULLs indefinitely at the maximum achievable rate. A compliant UUT will detect the TLI NULLs when it reaches the *ErrorWait* state. The UUT will reach the *Started* state at approximately time $T_{null}=T_0+19.2$ µs where it will begin transmitting NULLs before moving to the *Connecting* state. We define $T_{null}-T_0$ as the NULL arrival time.

Now consider what happens if the TLI sends another error at time $T_1>T_0$. If the UUT is in the *ErrorReset* state it will be unable to detect this error because its receiver is disabled. Thus errors received while the UUT is in *ErrorReset* will have no effect on the expected NULL arrival time of 19.2 µs. In contrast, if the UUT is in any other state when the error arrives then the UUT must return to *ErrorReset* and begin the link initialisation process again. This will add T_1-T_0 to the expected NULL arrival time. An example graph of the NULL arrival time against T_1-T_0 is shown Figure 2.

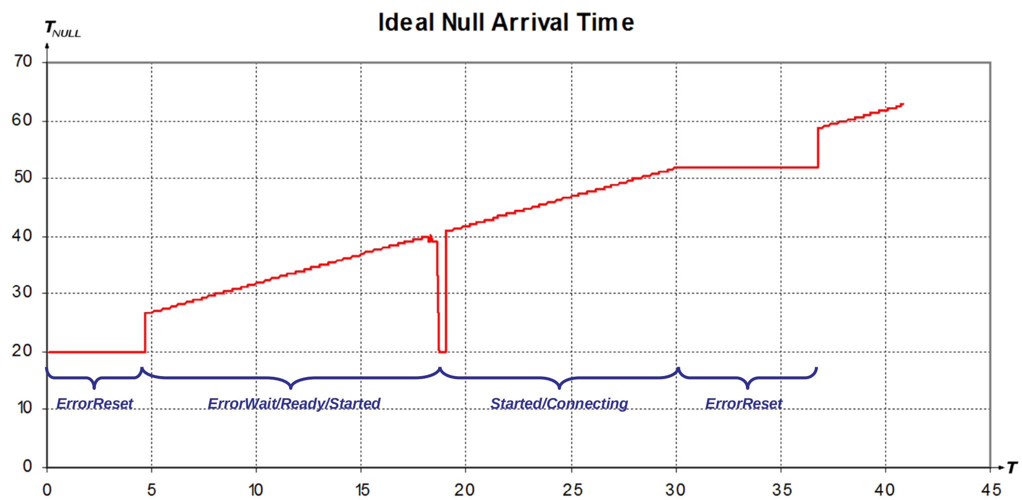

Figure 2: expected NULL arrival time graph

The horizontal segment of the graph from T=0 µs to T=4.64 µs corresponds to the period when the UUT is in the *ErrorReset* state: TLI errors are ignored and have no effect on the 19.8 µs NULL arrival time. The vertical step at T=4.64 µs marks the point where the UUT enters *ErrorWait* and the height of the step corresponds to the time the UUT spends in *ErrorReset* (approximately 6.4 µs). Note that the horizontal position of the vertical step is not 6.4 µs because the TLI waits until the UUT has disconnected before it starts the NULL arrival timer.

The diagonal section of the graph from T=4.64 µs to T=18.64 µs corresponds to the *ErrorWait*, *Ready* and *Started* states: if T_1 is increased by Δ then the NULL arrival time is expected to increase by Δ also producing a diagonal line. When the UUT is in the *Started* state its start-up NULLs will be detected by the TLI before the TLI is ready to issue the second error. In this situation the TLI waits until the UUT has disconnected after issuing the error at T_1 before enabling the start-up NULL detector. The dip in the middle of the graph corresponds to the period when the UUT has started sending NULLs in the *Started* state but the TLI has not detected them yet. The TLI sends the error at time T_1 and detects these NULLs shortly afterwards.

The end of the diagonal line at T=30.32 µs indicates where the UUT leaves the *Connecting* state because it hasn't received any FCTs from the TLI. The graph repeats at this point with another horizontal segment corresponding to *ErrorReset*.

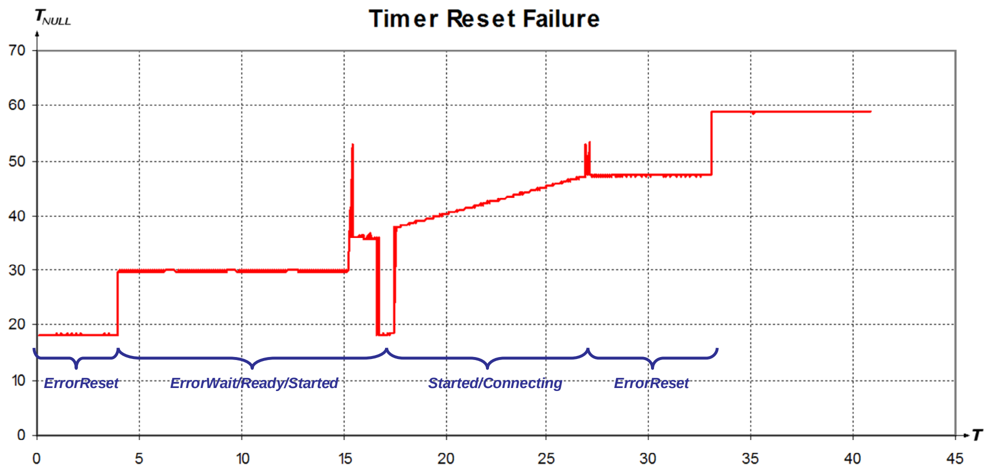

Figure 3: failure to reset 6.4 µs /12.8 µs timers

An example of NULL arrival times from a faulty device is shown in Figure 3. By comparing this graph with that shown in Figure 2 it can be seen that there is a problem with the *ErrorWait* state. It seems that the *ErrorWait* state is ignoring errors just like the *ErrorReset* state. However, since there is a step at T=3.96 µs the UUT appears to be detecting errors after leaving *ErrorReset*. This problem was due to a failure to reset a shared 6.4 µs/12.8 µs timer when moving from *ErrorWait* to *ErrorReset*. This caused the second *ErrorReset* period to be between 0 µs and 12.8 µs instead of 6.4 µs.

5 CONCLUSIONS

The Conformance Tester is very effective at detecting bugs in SpaceWire devices that were not uncovered by the development test suites of the devices. The development of the Conformance Tester has also raised questions about how parts of the SpaceWire standard ought to be interpreted such as the (10±1) Mb/s link initialisation rate measurement. Finally, the novel NULL arrival test described here can be used to probe the SpaceWire link initialisation state machine. It enables the 6.4 µs *ErrorReset* duration of any UUT to be measured and verifies the correct response to errors in the different link initialisation states.

6 REFERENCES

[1] "SpaceWire - Links, nodes, routers and networks", ECSS-E-50-12A, February 2003.