MEASURING TIME AND TIME-RELATED ASPECTS OF SPACEWIRE

Session: Test and Verification

Short Paper

Dr Barry M Cook, C Paul H Walker

4Links Limited, Bletchley Park, Milton Keynes MK3 6ZP, England

E-mail: [Barry, Paul] @4Links.co.uk

ABSTRACT

This paper presents a variety of measurement techniques that give information about how a device or system behaves with respect to time. We show how these techniques can be used to measure different parameters of SpaceWire, such as the receive link speed, the times of arrival of Time Codes, the durations and latencies of packets, and the operating margins of the SpaceWire receiver. One such operating margin is the disconnection timeout, and we report on the discovery of non-compliance with the ECSS SpaceWire standard on this parameter, and of other design errors that can be exposed by the measurement of time-related parameters.

We comment that asynchronous design can be difficult, that as a result its testing should be particularly rigorous, and that a synchronous SpaceWire receiver overcomes some of these issues.

We then show how the basic timing measurement techniques can be expanded to cover a potentially large SpaceWire system, and how the time-related issues of interface bandwidth and latency can be overcome to give tens of Gb/s overall SpaceWire test bandwidth.

1 INTRODUCTION

Users of our SpaceWire [1] test equipment have told us for many years how important time is in the design of their systems.

In response to these user needs we have developed a variety of techniques for measuring different time-related characteristics. We reported on some of this work at DASIA 2006 [2], and in this paper concentrate more on the how the measurements can be done and what they can be used for. Much of this is in helping users to remove bugs, but we have also found that our equipment exposes bugs that would not be seen without the measurement of time-related parameters.

We also show how the capabilities have been developed to cover systems from a single chip up to a large system with many ports and tens of Gb/s SpaceWire throughput.

The bugs that we have seen, including those in supposedly mature designs, lead us to sound a note of caution about asynchronous design and the traps that it can cause for the unwary.

2 MEASUREMENT TECHNIQUES

We present a variety of measurement techniques. All are simple in concept, but achieving the resolution, sensitivity and synchronization that users require is far from trivial.

Frequency: This is simply the inverse of time, and may be measured by timing a number of cycles of the signal that is being measured.

Time tagging: The occurrence of an event is sampled at high frequency, so that the time at which the event occurs can be recorded. The recording resolution on early products was 100ns, and this has been progressively improved to less than 1.5ns.

Fine control of transmit speed: Transmit speed can be set in small increments. The increments are 0.1Mb/s steps (or smaller) up to 40Mb/s, and 1Mb/s steps up to 400Mb/s.

Increased bit periods: Unlike most communication standards, a bit on SpaceWire can have any duration, and successive bits do not need to have the same duration. A particular bit duration may therefore be extended by a certain number of clock cycles.

Synchronization between outputs: A set of outputs can be delayed, without transitions, until all are ready to transmit, much as a barrier prevents horses from leaving the starting line in a race. When the barrier is lifted, all the packets leave at the same time, or with accurately pre-determined delays. Within a single EGSE box, this synchronization can typically be sub-nanosecond.

Synchronization between units: Some SpaceWire networks are being built with many SpaceWire nodes, and to test these, it is useful to have a common view of time across all the test equipment being used. Synchronization is achieved for time tags to ±3ns, over any number of boxes up to a total synchronizing cable length of 50metres.

3 MEASURING THE BIT RATES

It can be useful to be able to see not only that a link has connected, but also to see the transmit and receive bit rates. If the bit-rate is not as expected (whether as a result of a hardware or software failure or because the user has set it up incorrectly), it is not always obvious that this is the cause of unexpected behaviour. Displaying both the transmit and receive speeds on the front panel and making them available to software, for example to show on a GUI, makes such errors more readily visible.

The measurement is also useful for validation, as the ECSS SpaceWire standard requires that a link starts up between a rather narrow range of bit-rates and we have observed designs that start up at a transmit bit-rate outside this range.

4 MEASURING TIME CODES

SpaceWire uses Time Codes to distribute a common view of time throughout the spacecraft. Test equipment needs to ensure that the distributed time codes arrive correctly. Time tagging their arrival and logging the results makes it possible to see if any time codes get lost, to see what the time code interval is, and to see what long-term drift and short-term jitter there is in the time code source and distribution.

5 MEASURING PACKET DURATION

There are many places, both in an individual SpaceWire node and across a SpaceWire network, where packets can be delayed and so have Nulls inserted where they are not expected. Time tagging the start and end of packet to record the packet duration enables the user to determine whether such Nulls have been inserted.

6 MEASURING LATENCY OR DELAY

In a similar way to measuring packet duration, packet delay can be measured with time tags. In this case the transmitted packet is time tagged as well as the received packet. The measurement is so sensitive that it can easily discriminate differences in cable length, but is equally capable of measuring routing-switch or host-interface latency even if these are milliseconds or seconds. Indeed the measurements can be used at both ends of an Internet connection to measure local latency and remote latency, the difference being the round-trip-time over the Internet.

7 MEASURING OPERATING MARGINS

Many SpaceWire circuits operate at a limited number of transmit speeds. For example a design might run at 200Mb/s, 100Mb/s, 50Mb/s and 10Mb/s. The receiver may have been designed to operate at up to 220Mbits/s, but the circuit itself provides no means of testing such operation. Test equipment with the finely tuned transmit frequency can determine whether the design does indeed operate up to 220Mbits/s or whether it fails at a much lower speed and would therefore have inadequate design margin.

As well as being used to qualify completed designs, users have found this test capability extremely useful in development. One user had two equipments that worked individually but not together. It transpired that both had narrower than intended operating margins --- one only worked below a certain operating point, the other only worked above a certain operating point and there was no overlap between the two operating regions. Using the finely tuned transmit frequency, they were able to expand the operating margins so that they overlapped and the two equipments were able to work together.

8 MEASURING THE EFFECT OF HIGH-FREQUENCY NOISE

Asynchronous input circuits for SpaceWire can be susceptible to high-frequency noise. If the noise results in consecutive bits having less than the minimum edge separation that the receiver can tolerate, the regenerated clock may have inadequate pulse width. Behaviour in such circumstances is indeterminate and the receive logic may enter an invalid state. We have found several designs where the receiver gets into a state that can not be recovered by the normal disconnect and exchange of silence — the only way to recover was to reset the whole device.

Testing at deliberately higher speeds than the device can tolerate may seem to be excessive, but the high-speed testing is only simulating noise which may occur in flight as a result of radiation effects.

We would recommend:

- that this test should be performed on all asynchronous SpaceWire receivers;
- and that if asynchronous designs are used, then a mechanism is invoked that is able to reset the device if it gets into an invalid or otherwise irrecoverable state.

9 MEASURING DISCONNECTION TIMEOUT

The timing measurement techniques can be used to create a precise gap between transitions that is much longer than the normal bit-period, For example the gap can be set to 700ns, which should never result in a disconnection timeout, or to 1000ns, which should always result in a disconnection timeout. And of course a scan can be performed that tests at a range of gaps between edges, from less than 700ns to more than 1000ns. This can be done when the link is idle, just transmitting Nulls, or when it is transmitting data.

With one SpaceWire design, we found that a gap of 840ns never generated a disconnect timeout, and that a gap of 860ns always generated a timeout. With another SpaceWire design, we found that when receiving Nulls, it always generated a timeout at 1000ns, but when receiving data, a significant proportion of gaps above 1000ns failed to generate the timeout. This is not the sort of bug that would jeopardize a mission, but it is clearly not compliant with the ECSS SpaceWire standard.

10 TESTING ASYNCHRONOUS ARBITRATION

Synchronous arbiters are comparatively easy to test, and indeed if a simulation reports functional correctness and timing analysis reports that the required speed is met, confirmatory testing is possibly adequate.

The situation is very different in the case of asynchronous arbitration. Firstly, very few of the simulation tools adequately handle asynchronous circuits. And secondly, synchronous timing analysis is irrelevant when all the signals are asynchronous.

We therefore provide synchronization on packet outputs, with controllable offsets of one or more outputs with respect to the others. The synchronization is remarkable, typically below our capability of measuring it and definitely well under one nanosecond of skew between outputs. This is as close as possible to what is needed to test an asynchronous arbiter, for correct functionality let alone for performance.

One of the bugs found with this technique was a routing switch with an arbiter that was not ‘fair’, so it could allow some ports to block other ports permanently.

11 GENERAL COMMENTS ON ASYNCHRONOUS DESIGN

The last few sections have highlighted problems with different aspects of asynchronous design. While SpaceWire’s asynchronous nature has many advantages, it presents enticing traps for the unwary. The standard design tools are excellent for synchronous design but — except in expert hands — are almost non-existent for asynchronous design. Consequently those designers of SpaceWire who have chosen the asynchronous approach have given themselves an extremely challenging job.

Several years ago, 4Links abandoned asynchronous design and our receivers have a free-running clock that samples the asynchronous inputs at the device pins. Thereafter, the whole receiver is synchronous, and can be handled by the excellent design tools that are available. The result is a much more robust SpaceWire core than is possible with asynchronous design.

12 MEASURING TIME ACROSS AN ENTIRE SYSTEM

The above measurements have proved to be invaluable both for detecting and for guidance in fixing bugs. But as described so far, they are only available from a single test unit, which in our product family provides up to eight SpaceWire ports. SpaceWire systems are often smaller than this, but there are several spacecraft in design that have many more than eight SpaceWire ports. So it would be useful for time to be accurately coordinated across the system.

Synchronization has been achieved with a coax bus connecting all the SpaceWire test sets. The bus carries not only the synchronization clock, but also all the management of which unit is synchronization master (to which all the other units synchronize) together with calibration of the delay to each unit. The result is synchronization for time tags across any number of boxes up to a total coax length of 50metres to within ±3ns. Synchronized outputs are not quite so precise at ±20ns, but they can be time-tagged with the ±3ns precision.

Putting this into perspective, a time-tagged packet output on one unit can be received by a unit 50 metres away, with the synchronizing cable visiting ten or more other units, and with the ambient temperature changing and different at the two units — and the transit time reported by comparing time tags from the sending and receiving units is accurate to within less time than it takes for a signal to go along about half a metre of SpaceWire cable.

13 OVERCOMING BANDWIDTH LIMITATIONS

If both directions of a SpaceWire link are running at 400Mb/s, through eight ports per unit, the total bandwidth of actual data can be up to 5.12Gb/s. Add instrumentation, for example with time tags, and total bandwidth can increase to above 6Gb/s on a single unit.

To provide such bandwidth, we have extended our test equipment to include a Packet Generator for each port, which can be programmed to generate a long sequence of different packets with little or no interaction across the interface to a control computer. The computer just sets up the algorithm to generate the sequence of packets.

In the receive direction, the equipment has a packet checker that can use, for example, such information as the header and checksum to check that the received packet is correct. If it is correct, no (or negligible) information needs to be sent to the computer, and so bandwidth is conserved in this direction as well. In fact the data-rate multiplication in the generator and data-rate compression in the checker are such that full-bandwidth tests are conceivable over the Internet.

The packet generator and checker are actually implemented by small, highly specialized, processors — specialized for the jobs of generating and checking packets, with remarkable flexibility in so doing.

When the packet generator is combined with multiple units and their synchronized time-tags, systems can be tested with tens of Gb/s of SpaceWire traffic.

14 OVERCOMING LATENCY LIMITATIONS

Much work goes into overcoming bandwidth limitation but, in many cases, the actual bandwidth achieved is determined more by latency than by the native bandwidth limitations. This is particularly the case when a PC (with its operating system) has to respond to a request. The whole process can take many milliseconds.

Having a programmable packet generator and checker on each port, with the generator and checker able to transfer information between each other, provides an exceptionally low-latency response. The checker not only checks the packets it receives but extracts the information needed by the generator to return a packet with the right size and content to the correct place in the network. The response may be, for example, to a read request, or it might be a simple acknowledgement of the received packet.

As with the generator and checker working in isolation but over multiple units, responsive systems can be tested with tens of Gb/s of SpaceWire traffic.

15 CONCLUSIONS

Time is an important parameter for SpaceWire, and users need to know how their systems behave with respect to time. This includes being able to measure and record parameters such as the receive link speed, the times of arrival of Time Codes (and the variations in those arrival times), the durations and latencies of packets, and the operating margins of the SpaceWire receiver.

The asynchronous nature of SpaceWire, while offering many advantages, also offers enticing traps for the unwary. These traps can result in failure to comply with the ECSS SpaceWire standard, or failure in operation, for example as a result of a noise upset.

The tests that we have performed on a wide range of SpaceWire designs have exposed a variety of bugs. We observe that, even in the hands of experts, common design and simulation tools lead to a false sense of security and lead us to conclude that compliance with the standard can only be assured if a design has been subjected to the timing tests that we have outlined here.

If subjecting a SpaceWire design to noise causes a lock-up condition that can not be recovered by the standard disconnection and exchange of silence, we have to question whether the design is fit for purpose.

At least some of the problems we have uncovered are because asynchronous design is difficult. We and our users have found our own synchronous SpaceWire design to be robust.

If time is important over a single SpaceWire link, it is equally important across a large system with 100 or more SpaceWire nodes. Test equipment is now available that provides a consistent view of time from small systems to large systems, and to within a few nanoseconds.

Any interface to a large computer and operating system is likely to limit bandwidth and give excessive latency, distorting system behaviour with respect to time. Adding specialized packet processors at each port of the test equipment overcomes such limitations for many aspects of system test, and scales to tens of Gbits/s of SpaceWire traffic.

REFERENCES

[1] The ECSS-E-50-12 Working Group, "**ECSS-E-50-12A 24 January 2003**, SpaceWire - Links, nodes, routers and networks", published by the ECSS Secretariat, ESA-ESTEC, Requirements & Standards Division, Noordwijk, The Netherlands

[2] Barry M Cook, Paul Walker, "**Verifying the Temporal Behaviour of SpaceWire Components and Systems**", Data Systems In Aerospace (DASIA), Berlin, 22 to 25 May 2006, proceedings CD SP-630 – July 2006, ISBN 92-9092-941-3, ESA.