SPACEWIRE 2018

8TH INTERNATIONAL SPACEWIRE CONFERENCE 14TH-18TH MAY 2018 LOS ANGELES, USA

2018.SPACEWIRE-CONFERENCE.ORG















BACKGROUND IMAGE: ESA/HUBBLE & NASA; CC BY 4.0 WWW.ESA.INT/SPACEINIMAGES/IMAGES/2017/06/A_STORMY_STELLAR_NURSERY FOREGROUND IMAGE: NAVID SERRANO: CC BY-SA 3.0 HTtps://COMMONS.WIKIMEDIA.ORG/WIKI/FILE:LA_SKYLINE_MOUNTAINS2.

SpaceWire-2018

Proceedings of the 8th International SpaceWire Conference California, 2018

Editors: Steve Parkes and Carole Carrie





SpaceWire-2018

Proceedings of International SpaceWire Conference

California, 2018

ISBN: 978-0-9954530-1-2



© STAR-Dundee Ltd Dundee 2018

All rights reserved. No part of this publication may be reproduced or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of the publisher.

Preface

These proceedings contain the papers presented at the 2018 International SpaceWire Conference, held in Long Beach, California, USA between 14th and 17th May 2018. The International SpaceWire Conference brings together international spacecraft engineers and academics who are working on spacecraft on-board data-handling technology. It is of benefit to product designers, hardware engineers, software engineers, system developers and mission specialists interested in and working with SpaceWire, enabling them to share the latest ideas and developments related to SpaceWire and SpaceFibre technologies.

SpaceWire is now being used or designed into well over one hundred spacecraft, covering science, exploration, Earth observation and commercial applications. High profile missions like James Webb Space Telescope, GAIA, ExoMars, Bepicolombo, Sentinels 1, 2, 3 and 5 precursor, and GOES-R are using SpaceWire extensively. SpaceWire is being used in Europe, Japan, USA, Russia, China, India, and other countries of the World.

SpaceFibre is the next generation of SpaceWire technology, offering higher data-rates and substantially enhancing the capabilities of SpaceWire. It runs over electrical or fibre optic cable covering distances of 5m and 100 m respectively while running at data rates of up to 3.125 Gbits/s with 6.25 Gbits/s currently under development in radiation tolerant technology. SpaceFibre incorporates quality of service, providing multiple independent virtual networks for transferring information over the physical network, each virtual network having its own priority, bandwidth allocation and schedule. These capabilities enable SpaceFibre to provide deterministic data delivery without loss of network bandwidth for combined control and payload data-handling networks. It also provides integrated, rapid fault detection, isolation and recovery technology, which makes SpaceFibre a highly robust network for use in applications where reliability and availability are critical. SpaceFibre multi-laning technology extends the bandwidth of a link using multiple lanes and provides hot and cold redundancy and graceful degradation. Asymmetric links are also possible, using uni-directional lanes provided at least one lane is bi-directional. SpaceFibre uses the same packet format and addressing concepts as SpaceWire making it trivial to connect existing SpaceWire equipment into a SpaceFibre network. IP cores for radiation tolerant FPGAs and ASICs with SpaceFibre interfaces are available and under development. A wide range of test and development equipment is now available.

The conference covers many different aspects of SpaceWire and SpaceFibre technology and includes both academic and industrial presentations. Sessions address recent developments of the SpaceWire set of standards, space missions and other applications using SpaceWire, new components, sensors and cables which support the SpaceWire standard; products supporting SpaceWire including onboard equipment, instruments and related onboard software; methods and equipment to aid the test and verification of SpaceWire components, units and systems; and SpaceWire networks, their architecture, configuration, and discovery, as well as higher level protocols and related hardware and software design issues. The new sessions on SpaceFibre illustrate how this next generation of SpaceWire technology is gaining momentum, already being designed into spaceflight systems. It is an exciting time in the SpaceWire community as this latest technology literally begins to take off.

A tutorial session gives a detailed technical introduction to SpaceFibre, describing its potential applications and explaining how the key features of SpaceFibre operate.

The community of engineers working on SpaceWire meet regularly at the SpaceWire Working Group meetings to help with the further development of SpaceWire, SpaceFibre and related standards and technologies. This group includes engineers from many parts of the World with substantial contributions from Europe, USA, Japan, and Russia. The SpaceWire Conference complements these Working Group meetings with more formal presentations from a wider range of contributors. The conference committee would like to acknowledge the support and hard work of the many individuals who made International SpaceWire Conference 2018 a reality. First, we thank the authors and the keynote speakers for their high-quality contributions. We express our gratitude to the Technical Committee for their assistance in the review process. We thank all people supporting us at Microsemi, the Space Technology Centre, University of Dundee, and the European Space Agency. Finally, we would like to give a special thanks to Carole Carrie at STAR-Dundee for making the conference happen.

The Conference Chairpersons,

James Lux, Jet Propulsion Laboratory, USA Ken O'Neill, Microsemi Corporation, USA Steve Parkes, Space Technology Centre, University of Dundee, UK Dirk Thurness, European Space Agency, The Netherlands

Technical Committee

Susan Clancy – NASA JPL, USA

Brian Cox - NASA JPL, USA

Brice Dellandrea - Thales Alenia, France

Seisuke Fukuda – JAXA/ISAS, Japan

Wahida Gasti - ESA, The Netherlands

Sandi Habinc – Cobham Gaisler, Sweden

Hiroki Hihara - NEC, Japan

Christophe Honvault - ESA, The Netherlands

Torbjörn Hult - RUAG Space, Sweden

Jørgen Ilstad - ESA, The Netherlands

Paul Jaffe - Naval Research Laboratory, USA

David Jameux- ESA, The Netherlands

Clifford Kimmery - USA

Alexander Kisin - NASA GSFC, USA

Robert Klar - South West Research Institute, USA

Jim Lux - NASA JPL, USA

Giorgio Magistrati - ESA, The Netherlands

Keiichi Matsuzaki - JAXA, Japan

Minoru Nakamura - Mitsubishi Electric, Japan

Masaharu Nomachi - University of Osaka, Japan

Olivier Notebaert - Astrium SAS, France

Steve Parkes - University of Dundee, Scotland, UK

Paul Rastetter - Astrium GmbH, Germany

Alan Senior - Thales Alenia Space UK Ltd, UK

Yuriy Sheynin - St. Petersburg State University of Aerospace Instrumentation, Russia

Felix Siegle - ESA, The Netherlands

Tatiana Solokhina - ELVEES, Russia

Antonis Tavoularis – ESA, The Netherlands

Dirk Thurnes - ESA, The Netherlands

James Windsor - ESA, The Netherlands

Takahiro Yamada - JAXA/ISAS, Japan

Monday 14 May

- 13:00 14:30 Registration
- 14:00 18:00 SpaceFibre Tutorial

Tuesday 15 May

- 08:00 16:30 Registration
- 09:00 10:30 Conference Opening / Keynote Presentations (90 min)
- 10:30 11:00 Test & Verification Short (30 min)
- 11:30-12:35 Missions & Applications Long/Short (65 min)
- 14:05 14:55 Components 1 Long (50 min)
- 15:25 16:15 Components 2 Long (50 min)

Wednesday 16 May

- 08:30-11:30 Registration
- 09:00 10:45 Components Short (105 min)
- 11:15 12:00 On-board Equipment Short (45 min)
- 13:30 15:00 Networks & Protocols Short (90 min)
- 15:00 16:30 Poster Session (90 min)

Thursday 17 May

- 08:30-11:00 Registration
- 09:00 10:15 On-board Equipment Long (75 min)
- 10:45 12:25 Networks & Protocols Long (100 min)
- 13:55 14:45 Test & Verification 1 Long (50 min)
- 15:15 16:30 Test & Verification 2 Long (75 min)

Programme is subject to change

Tuesday 15 May

Test & Verification (Short)

Testing over Ethernet with the SpaceWire GbE Brick

Test and Verification, Short Paper

Stuart Mills, Chris McClements, David Paterson, Pete Scott STAR-Dundee Ltd. Dundee, Scotland, UK

Abstract— This paper introduces a new SpaceWire test and development product with a Gigabit Ethernet interface developed by STAR-Dundee, the SpaceWire GbE Brick. It describes the unique features of the SpaceWire GbE Brick which make it ideal for both remote use in a test chamber and when connected directly to a PC using an Ethernet cable. The paper also characterises the performance of the SpaceWire GbE Brick and compares the performance to SpaceWire devices with PCI and USB interfaces.

Index Terms—SpaceWire, Gigabit Ethernet, GbE, Brick, SpaceWire GbE Brick.

I. INTRODUCTION

A common use case for SpaceWire test and development equipment is to locate it in a test chamber and then access it remotely to test a SpaceWire device or network.

One solution to this problem is to include a PC in the test chamber to which the SpaceWire test and development equipment is connected. The user can then remotely connect to this PC over an Ethernet network. This has the advantage that the PC in the test chamber transmits and receives any SpaceWire data and can therefore do so at high throughput rates and with low latency. The remote PC may only need access to the transmission and reception statistics, so high throughput and low latency may not be required between the two PCs.

This solution is not very elegant, however, and is not always feasible, e.g. where space is limited or when the PC's presence may affect the test. An alternative solution would be to include the interface to the Ethernet network within the SpaceWire test and development equipment. To ensure that the unit could be used in test chambers where there is not enough space for a PC, it's essential that such a device be small. This gives the added benefit of galvanic isolation between the test PC and the SpaceWire network. As SpaceWire traffic may be transmitted to the device from a remote PC, it must also include a high-speed interface and efficient software capable of transmitting and receiving at high data rates over that interface.

To meet these requirements, a new test and development product with a Gigabit Ethernet (GbE) interface has been developed by STAR-Dundee, the SpaceWire GbE Brick. This will begin shipping to customers in the third quarter of 2018 and is shown in Fig. 1. Steve Parkes Space Technology Centre University of Dundee Dundee, Scotland, UK



Fig. 1. SpaceWire GbE Brick

The SpaceWire GbE Brick can act as a SpaceWire interface or a router, it has two SpaceWire ports which can operate at link speeds of up to 400 Mbits/s and has two external triggers, each of which can operate as an input or an output. It is housed in a small case measuring around 85mm on its longest side. The software interface is the same as for other STAR-Dundee interface and router devices, making it possible to use software developed for these other products with the GbE Brick. This software, called STAR-System, offers high throughput and low latency when transmitting and receiving packets and has been updated to support high performance over Gigabit Ethernet.

This paper describes the unique features of the SpaceWire GbE Brick which make it ideal, not only for remote use in a test chamber, but also when connected directly to a PC using an Ethernet cable. The paper also characterises the performance of the SpaceWire GbE Brick and compares it to SpaceWire devices with PCI and USB interfaces.

II. FEATURES OF THE SPACEWIRE GBE BRICK

The SpaceWire GbE Brick includes all the features provided by its USB equivalent, the SpaceWire-USB Brick Mk3 [1], [2]. The SpaceWire GbE Brick is externally powered and has two SpaceWire ports which can be accessed from software using independent channels, so traffic on one port is not affected by traffic on the other port. A third independent channel provides access to the device's configuration port, allowing configuration of the device while concurrently transmitting and receiving on the two SpaceWire ports. In addition to the default interface mode, the device can operate as a full SpaceWire router, supporting both path and logical addressing.



Fig. 2. Displaying received images in the STAR-System Sink application

The device can act as a time-code master, and can both transmit and receive individual time-codes. Error injection capabilities are provided to inject several different error types on the link. As with all STAR-Dundee products, support is provided and upgrades are offered to add new features and fix issues at no additional cost.

The device can be field upgraded using a graphical user interface (GUI) application included in the STAR-System software suite, which is also included with the product. Version 4.0 of STAR-System is being released in conjunction with the SpaceWire GbE Brick. This release supports the latest Linux kernels and Windows releases and is backwards compatible with all previous releases of STAR-System. It provides a consistent interface for all STAR-Dundee interface and router products released in recent years. This means that software developed for the SpaceWire-USB Brick Mk3 or the SpaceWire PXI Interface [3], for example, can be used with the SpaceWire GbE Brick. Both C and C++ APIs are provided for interacting with STAR-System, along with comprehensive documentation and examples provided as source code, which can be used as a basis for new applications.

The APIs can easily be integrated in to other programming languages, with users known to be developing with the APIs under C#, Java and Python. A LabVIEW wrapper [4] is also available separately for using the SpaceWire GbE Brick and other STAR-System devices from National Instrument's LabVIEW visual programming language.

A key feature of the APIs is that they not only provide functionality to transmit and receive packets, but also functions required when testing SpaceWire equipment. For example, the APIs make it simple to transmit packets terminated with an EEP, and to determine the end of packet marker of received packets. An RMAP Packet Library is also included, which provides functions for creating Remote Memory Access Protocol (RMAP) [5] packets to be transmitted, checking the validity of received RMAP packets, and obtaining the values of fields in an RMAP packet.

Among the many new features included in version 4.0 of STAR-System is the ability to transmit images from a webcam using the Source application and to display images received in the Sink application, illustrated in Fig. 2. These features can be useful to simulate a camera on the network or to view data from a SpaceWire or SpaceFibre Camera [6].

III. PERFORMANCE OF THE SPACEWIRE GBE BRICK

A concern with the use of Ethernet to interface to a SpaceWire device is the poor latency in comparison to buses used by other STAR-Dundee devices, such as USB and PCIe. Although Gigabit Ethernet operates at a line rate of 1 Gbit/s, throughput is also a concern when using the device on busy Ethernet networks.

To achieve the best performance, STAR-System has been updated in version 4.0 to support Gigabit Ethernet devices, with considerable effort made to achieve the best possible throughput and latency. To characterise this performance, the SpaceWire GbE Brick was tested and compared with other STAR-Dundee products which use alternative bus technologies.

TABLE I. shows information about the bus technology used by each of the products tested and the number of SpaceWire ports on that device. Most of the buses used by these products are shared buses, meaning that performance will be affected by other traffic on the bus. While this may not be an issue for the SpaceWire PCI Mk2 [7], for example, if it is the only PCI device in a PC, it's likely to be an issue for the SpaceWire GbE Brick when used on a busy Ethernet network.

Product	Bus	Shared Bus?	Bus Line Rate	SpW Ports
SpaceWire-USB Brick Mk2 [8]	USB 2.0 (High Speed)	Yes	480 Mbits/s (half-duplex)	2
SpaceWire-USB Brick Mk3 [2]	USB 3.0 (SuperSpeed)	Yes	5 Gbits/s	2
SpaceWire GbE Brick	Gigabit Ethernet	Yes	1 Gbit/s	2
SpaceWire PCI Mk2 [7]	PCI	Yes	1,064 Mbits/s	3
SpaceWire PCIe [9]	Single Lane PCIe 1.1	No	2 Gbits/s	3

TABLE I. PRODUCTS UNDER TEST AND THEIR BUS PROPERTIES

Fig. 3. shows the results of latency tests for these products. The latency tests involve transmitting a packet from the STAR-Dundee Performance Tester application to the device under test, which sends the packet over a SpaceWire link operating at 200 Mbits/s. This link is looped back to the transmitting device. The packet is then passed back to the Performance Tester application where it is received. This sequence is repeated 1,000 times and the average latency calculated from the total time taken. The values shown in the chart in microseconds on the y-axis are therefore the average times that it takes to transmit a packet from the application. The test is repeated for packet sizes from 1 to 1,000 bytes, shown along the x-axis.

The source code and executable for the Performance Tester are included with STAR-System so that users can perform the tests themselves while also providing an example of how to get the best performance from the STAR-System APIs. The test results shown in the chart were all performed on a Windows 7 PC with an Intel Core i7-4790 processor and 12 GBytes of RAM. This is a 3-year-old PC with a relatively fast processor when it was released, so should be representative of many PCs in common use at the time of writing. The SpaceWire GbE Brick tests were performed with the device connected directly to the PC, meaning that no Ethernet switches or hubs were crossed and there was no traffic from other devices on the shared bus.

These results show that the latency achieved by the SpaceWire GbE Brick is generally quite good, but is occasionally very poor in comparison to the other products. It is assumed that these sporadic increases in latency are due to the PC attempting to perform other operations on the network. The results when repeating the tests support this theory, with the peaks appearing at different packet sizes.

The Performance Tester application also provides tests for measuring the throughput of devices. These tests transmit and receive packets at very high rates and measure the rate at which the packets are received. The results of these tests are shown in Fig. 4. This chart shows that the variances in latency do not affect average throughput. Performance of the SpaceWire GbE Brick is very similar to the other products, and is around the maximum that can be achieved on a 200 Mbits/s link. Note that the performance of the SpaceWire-USB Brick Mk2 [8], which connects to the PC using USB 2.0, is worse than expected. It was found that this is the result of using a USB 2.0 device with a USB 3.0 controller. Using the device on an older PC with a USB 2.0 controller, the performance was similar to the other products, despite the PC having a lower specification.



Fig. 3. Latency test results over SpaceWire at 200 Mbits/s for several STAR-Dundee products



Fig. 4. Throughput test results over SpaceWire at 200 Mbits/s for several STAR-Dundee products

While these tests show the performance for one link transmitting and receiving at a SpaceWire line rate of 200 Mbits/s, it should be remembered that the devices being tested have more than one link, each link can operate at faster line rates than 200 Mbits/s, and configuration commands may also be sent to the device while transmitting and receiving. It should also be remembered that the tests are being performed with the SpaceWire GbE Brick connected directly to the PC, rather than on an Ethernet network.

To compare performance when the device is accessed over an Ethernet network, tests were run over STAR-Dundee's Star House office network. The latency tests recorded in Fig. 5. were performed with the controlling PC in a top floor office and the device connected to an Ethernet port in a basement laboratory. Traffic between the PC and the SpaceWire GbE Brick therefore crossed two Ethernet switches. The chart also includes the previous latency figures achieved when the device was connected to the PC for comparison.

Although average latency over the Ethernet network is comparable to when connected directly, this chart also shows that at times latency can be much worse, with some tests encountering average latencies of around 1.5 milliseconds. As with the discrepancies in the earlier latency test, this can be attributed to other traffic on the Ethernet network.

To test the maximum throughput rates that can be achieved, and whether these rates are high enough to accommodate two links transmitting and receiving while the device is also being configured, throughput tests were performed using the internal SpaceWire router. These tests do not use the SpaceWire links, so are not limited by the link speed. Two tests were run in parallel so that data was being transmitted from channel 1 of the device to channel 2, while concurrently transmitting in the opposite direction. Fig. 6. shows the results of these tests when performed over the Star House network and when the SpaceWire GbE Brick was connected directly to the test PC. Results are also provided for the Brick Mk3 for comparison.

Although these tests show that performance of the SpaceWire-USB Brick Mk3 is better than for the SpaceWire GbE Brick, the GbE Brick should be capable of keeping two links relatively busy in both directions at the maximum link speed of 400 Mbits/s. For a 400 Mbits/s link, the maximum theoretical data rate is 320 Mbits/s due to each 8-bit byte being encoded as 10 bits. This is further reduced when transmitting in both directions on a link, as flow control tokens must be transmitted along with the data, reducing the theoretical maximum to around 304 Mbits/s. With two links, this equates to a data rate of around 608 Mbit/s, while the SpaceWire GbE Brick can manage a total data rate of around 490 Mbits/s.

This chart also illustrates the effect of other traffic on the network. When connected directly to the PC, the average data rate of the SpaceWire GbE Brick occasionally drops by around 50 Mbits/s. When the device is accessed across the Star House network, the average rate drops by as much as 130 Mbits/s.

One further point to note on these tests is that the CPU usage for the SpaceWire GbE Brick tests (between 60% and 27%) was considerably higher than for the SpaceWire-USB Brick Mk3 tests (between 37% and 10%). It is possible that further work can be performed to optimise this CPU usage within STAR-System for Ethernet devices, but the additional overheads of TCP/IP may mean that it is not possible to attain the same low CPU usage of the USB and PCI devices.

IV. DEALING WITH THROUGHPUT AND LATENCY LIMITATIONS

While the SpaceWire GbE Brick delivers adequate throughput and latency performance for most applications, it's clear from the STAR-System Performance Tester results that when operating over an Ethernet network, latency and, as a result, throughput can be affected.



Fig. 5. Latency test results for SpaceWire GbE Brick connected locally and over an Ethernet network

The SpaceWire GbE Brick has been designed to limit the effects of these, however. Buffering is provided in the device to reduce the effect of jitter in the latency when transmitting and receiving. This has the effect of smoothing out the rate at which traffic is transmitted over the SpaceWire link, and limiting any delay on the SpaceWire link when receiving traffic. This explains why the changeable latency does not affect the throughput rates shown in Fig. 4.

For applications which require deterministic behaviour, the device includes triggering functionality which is supported by STAR-System's Triggering API. This allows actions to be performed when specific events occur. These events could be a

packet being received, an error being encountered, a pulse on the device's external trigger port, a time period elapsing or several other events. Possible actions include to transmit a packet or a time-code, pulse the device's external trigger or inject an error.

These combinations of events and actions allow deterministic behaviour to be implemented in user applications which may be running on a PC at the other side of the Ethernet network on which the SpaceWire GbE Brick is connected. Example applications are to transmit packets periodically, to transmit packets when a time-code is received and to inject errors when an external trigger is received.



Fig. 6. Double loopback throughput test results using internal SpaceWire router

V. CONCLUSIONS

This paper has presented a powerful new SpaceWire test and development product, the SpaceWire GbE Brick, which allows SpaceWire tests to be performed remotely over an Ethernet network. The SpaceWire GbE Brick and accompanying STAR-System software have been designed to cope with the effects of operating on an Ethernet network, which can affect the performance when transmitting and receiving packets. It has been shown that performance of the SpaceWire GbE Brick is comparable to that of STAR-Dundee's other test and development products.

Furthermore, through use of STAR-System's Triggering API it is possible to circumvent any potential latency jitter on an Ethernet network and achieve deterministic results, by performing actions based on events such as timers.

The SpaceWire GbE Brick is currently in production and will begin shipping to customers in the third quarter of 2018.

ACKNOWLEDGEMENTS

STAR-Dundee would like to acknowledge the feedback provided by the users of its products, which act as requirements for improvements to existing products and features of new products.

REFERENCES

[1] S. Mills, P. Scott, S. Parkes, "High Speed Test and Development with the SpaceWire Brick Mk3", International SpaceWire Conference 2014, Athens, Greece, September 2014.

- STAR-Dundee, "SpaceWire-USB Brick Mk3", <u>https://www.stardundee.com/products/spacewire-usb-brick-mk3</u>, STAR-Dundee Website, accessed April 2018.
- [3] STAR-Dundee, "SpaceWire PXI", <u>https://www.star-</u> <u>dundee.com/products/spacewire-pxi</u>, STAR-Dundee Website, accessed April 2018.
- [4] STAR-Dundee, "STAR-System for LabVIEW", <u>https://www.star-dundee.com/products/star-system-labview</u>, STAR-Dundee Website, accessed April 2018.
- [5] ECSS, "SpaceWire Remote memory access protocol", Standard ECSS-E-ST-50-52C, Issue 1, European Cooperation for Space Standardization, February 2010.
- [6] S. Parkes, A Srivastava, C. McClements, P. Scott, D. Dillon, A. Ferrer Florit, A. Gonzalez Villafranca, "SpaceFibre Camera", International SpaceWire Conference 2018, Long Beach, USA, May 2018.
- [7] STAR-Dundee, "SpaceWire PCI Mk2", <u>https://www.star-dundee.com/products/spacewire-pci-mk2</u>, STAR-Dundee Website, accessed April 2018.
- [8] S. Mills, A. Mason, C. McClements, D. Paterson, I. Martin, S. Parkes, "Developing SpaceWire Devices with STAR-Dundee Test and Development Equipment", International SpaceWire Conference 2013, Gothenburg, Sweden, June 2013.
- [9] STAR-Dundee, "SpaceWire PCIe", <u>https://www.star-dundee.com/products/spacewire-pcie</u>, STAR-Dundee Website, accessed April 2018.

Debug and Verification of SpaceWire links

Test and Verification Session, Short Paper

Daniel DeLazari, Alexsander Deucher, Angela Santos, Saulo Finco CITAR Project - CTI Campinas, Brazil

Abstract—The purpose of this paper is to show debug and verification methods of SpaceWire communication links using an oscilloscope. A new decode algorithm for continuous SpaceWire data streams will be discussed in combination with powerful trigger features. SpW communication data can be easily crosscorrelated with power and digital events on the satellite. Skew between SpW Data and Strobe is a severe threat to link operation. By making use of a variety of oscilloscope functions, the amount of link skew is evaluated in different links. Further, conformance test capabilities such as eye diagram will be introduced. The paper proposes to use a fast hardware-based clock recovery on the data signal to create an eye diagram.

Index Terms — SpaceWire, Physical Layer, Data Strobe Skew, Eye Diagram, SpaceWire Decode.

I. INTRODUCTION

When debugging a SpaceWire communication link, the problem may reside in different levels of the protocol stack. So, the test engineer must analyze the communication at different levels as well. It is not so easy to determine where the problem is. If the issue seems to be on the Physical Layer, the most common analysis tool is an oscilloscope.

Figure 1 shows the SpaceWire Protocol Stack. With most oscilloscopes, it is not possible to get SpaceWire decoded data, but only transition time, pulse widths, propagation delay, and so on.



Fig. 1. SpaceWire Protocol Stack

Armin Horn¹, Matthias Beer², Volker Ohlen² Test & Measurement Division Rohde & Schwarz Beaverton, USA^{1/2} - Munich, Germany²

The debug task becomes even more difficult if the problem is not only on the Physical Layer. This paper presents an oscilloscope which covers more than the Physical Layer. Since, besides basic signal measurements, it can also extract eye diagrams from SpaceWire signals, decode SpaceWire data, and also trigger on a continuous SpaceWire data stream.

II. SPACEWIRE TRIGGER AND DECODE

The decode feature allows the display of the logical content from a captured signal. Unlike protocol analyzers or network instruments, oscilloscopes are only able to decode the currently captured content in the acquisition window. Without the information of a communication handshake, the decoder option on an oscilloscope must therefore find other ways to identify packets in the data stream.

SpaceWire has no easily identifiable packet structure, as packets are not separated by regular spaces or marked by specific data patterns. The Rohde&Schwarz SpaceWire decoder uses the parity bits to align the packets, using an elimination algorithm. In order for this algorithm to work though, at least 110 bits of error free data are required at the beginning of the acquisition window. Once aligned, the decoder is able to identify all SpaceWire packet types and present them to the user.

A. Graphical Display

A very intuitive way to display the data is a graphical display on top of the respective traces. By aligning the start and stop of each packet with the signal, it is possible for the user to see which high/low transitions in the data stream are part of a particular package. This is specifically helpful when the signal shows errors due to signal degradation or other errors, for example glitches.

The decoder identifies and marks such errors. In the case of SpaceWire, this will typically be parity errors. Using the zoom functionality, the user can now localize and investigate these issues.

B. Result Table Display

Especially for large acquisitions, displaying the identified packets in a result table can be more helpful. The result table lists all the decoded packets with start/stop time, (error)-status and value.

For automation, or advanced data processing, it is possible to query this result table using remote command or simply exporting the table into a common file format.

C. Search Functionality

SpaceWire has many small packets. A large acquisition window can easily produce many hundred decoded packets. Using the search functionality, the user can filter the interesting results and ignore others.

D. Trigger Functionality

In a signal stream, how can the user even identify an interesting area? For example, when do certain rare events happen? These can be errors, or simply the start or end of a data transmission.

This is where a decoder based trigger functionality is helpful. Using a basic edge trigger, this SpaceWire trigger functionality identifies any bus activity, decodes the data and then searches them for the required events. This process will be repeated until at least one event has been detected. The localized event is then displayed at a trigger marker. Figure 2 shows an example of EOP Trigger, which is the end of a packet (seen just after the screen center).



Fig. 2. Graphical Decoder Display showing a Trigger on EOP

Possible SpaceWire character triggers are:

- CONTROL FCT, EEP, EOP, or ANY
- DATA any value (equal, not equal or range)
- NULL
- TIMECODE any value (equal, not equal or range)
- ERRORs parity or ESCAPE errors

III. SPACEWIRE DATA-STROBE SKEW

SpaceWire successful operation at high bit rates is very much dependent on the skew between Data-Strobe pair on the entire network. That is the case, since SpaceWire protocol relies on a minimum amount of time between any 2 edges of data and strobe signals for correct decode.

In order to ensure that, the SpaceWire standard determines that the skew should be minimized during the entire path between transmitter and receiver, which includes:

Data/Strobe CMOS generation/propagation (source logic)

- LVDS driver
- PCB tracks between LVDS driver and SpW connector
- SpaceWire cable (and connectors)
- PCB tracks between SpW connector and LVDS receiver
- LVDS receiver
- CMOS propagation and usage by destination logic

Annex 1 of SpW Rev 1 [1] standard describes how to perform some "isolated" measurements, like driver only, receiver only, cable only, and so on. That is only possible if you have access to all individual components/parts.

The approach in this paper was to use the entire link (driver, connectors, cables, and receiver) in order to have the sum of data/strobe skew in that particular link.

In order to do that, 9 different devices were used in this section. Skew was checked in 41 different point to point links between these 9 devices. Different cable arrangements were used (1m, 1m+1m, 1m+2m) to connect the 2 SpW devices.

This paper compares 2 techniques to automatically measure SKEW between SpW Data and Strobe:

- 1. Using scope built-in measurements of SETUP and HOLD
- 2. Creating a MATH function, which is an XOR of DATA and STROBE. Then measure duration of HIGH PULSE on this math function.

As shown in Figure 3, in the case of SETUP/HOLD, clock slope is set to "Either" [2]. This way, scope is measuring time distance between any transition on "clock" to any transition on "data". Setup considers data change first, and hold considers clock changes first. It is irrelevant if SpaceWire Data is assigned to scope clock or data. In both cases, time distance between any edge of STROBE to any edge to DATA will be measured (and the opposite too).



Fig. 3. Scope setting for SETUP/HOLD measurements

In the case of MATH XOR function between channels with DATA and STROBE, the high (positive) or low (negative) pulse could be measured, which also indicates the distance between DATA and STROBE edges in "any direction".

Measurements STATISTICS were enabled in both cases (SETUP/HOLD and MATH XOR), so that we can detect the maximum value that had occurred in a period of time. For better statistics, we wait for scope to take at least 10000 events.

On Figure 4, there are measurements for Setup and Hold and Positive Pulse on Math XOR. One set is for channels 1 and 2 and another set is for channels 3 and 4. There were 11454 events in this particular case.



Fig. 4. SpW Data-Strobe SKEW measurements with scope

T1 and T6 are LVDS transceivers. As it can be seen on Fig. 5, channels 1 and 2 have signals from T6 driver and channels 3 and 4 have signals coming from T1. Between them, there is 1m SpaceWire cable. In this example, both measurements were done at T6 side (or close to T6 driver/receiver).



Fig. 5. Physical setup for SKEW measurements

Table I shows SETUP/HOLD and MATH XOR results for T6/T1 communication pair like shown on Fig. 5.

			SETUP HOLD		Math XOR	
Scope Channel	LVDS XCVR	Setup Max	Hold Max	Max Setup Hold	H pulse	Scheme Difference
		(ns)	(ns)	(ns)	(ns)	(ns)
Ch 1 - 2	Т6	5.02	5.35	5.35	5.5	0.15
Ch 3 - 4	T1	5.19	5.42	5.42	5.7	0.28

TABLE I. SETUP/HOLD AND XOR COMPARISON

The measured skew is about 0.5ns for channels 1 and 2 and 0.7ns for channels 3 and 4. The difference between values got from "Setup/Hold approach" and "Math XOR approach" was 0.15ns for channels 1 and 2 and 0.28ns for channels 3 and 4.

On table II, we can see the comparison between the 2 approaches used on all 41 point to point links. The table shows

the difference between skew measured with Setup/Hold and Math/XOR approaches. In this case:

- 9 devices were used, and all links were at 200Mbps
- measurements were done at the receiver side
- measurements were done with different cable arrangements (1m, 1m+1m, 1m+2m)

Scheme Difference	1m cable	3m cable
MAX	0.40 ns	0.65 ns
AVG	0.16 ns	0.20 ns
Std Dev	0.11 ns	0.19 ns

TABLE II. SKEW APPROACH DIFFERENCE IN 41 LINKS

Taking into account all 41 links measured, the maximum difference between the 2 approaches was 0.4ns for 1m SpW cable and 0.65ns for 3m SpW cable.

That can be considered small considering probes bandwidth. Therefore, we can conclude both methods were equivalent to detect SKEW between DATA and STROBE in a SpaceWire link.

IV. SPACEWIRE EYE DIAGRAMS

Eye diagrams are very important to determine the quality of many high speed communication links. SpaceWire is no different in this aspect. Besides reflecting all kinds of skew present in the link components, eye diagrams also give an idea about amount of jitter present. For this section measurements:

- 10 devices were used in different combinations
- used scope EYE measurement (autoset)
- measured Height(mV), Width(ns) and Q factor
- used 200Mbps when possible. One device (T4) is limited to 100Mbps, and another is limited to 40Mbps
- measured with NO cable (close to driver and receiver), but also with different cable arrangements (1m, 1m+1m, 1m+2m)

Figure 6 shows SpaceWire EYE diagrams between two LVDS transceivers T1 using a 3m SpaceWire cable.



Fig. 6. EYE diagram - T1 to T1 with 3m cable (measured at RX side)

Table 3 shows eye measured values for 3 different setups.

Setup	Eye height	Eye Width	Q factor
3m SpW cable	443 mV	4.2 ns	28
1m SpW cable	545 mV	4.4 ns	37
NO cable	593 mV	4.7 ns	47

TABLE III. EYE VALUES - T1 TO T1 IN 3 SETUPS

Hardware based clock-data recovery allow more eyes processed than conventional methods. In addition to eye measurements like described before, it is also possible to setup a MASK inside the eye. Masks are used to determine whether the signal remains within specified limits. Therefore any violations can be seen or statistics on violation can be determined.

V. SPW ACTIVITY CORRELATED WITH CURRENT VALUES

In this section, we've used R&S RT-ZVC02 Multi-Channel Power Probe to measure current at 2 different points. In some cases, it is useful to correlate signals and messages with power consumption. This probe allow us to perform 2 current and 2 voltage measurements in parallel with the 4 differential analog probes.

Figure 7 shows a SpaceWire disconnect where there is "exchange of silence" as per protocol state machine. Besides the 2 SpaceWire signals for 2 devices, we also have 2 current measurements, with a total of 6 signals.

On example below, current on Z1.I1 dropped from 34.6 to 31.4mA, and current on Z1.I2 dropped from 11.1 to 7.9mA.



Fig. 7. CURRENT drop during SpW DISCONNECT

Using ZOOM functionality, details on 4 signal waveforms can be observed. It is also possible to enable DECODE on both SpaceWire sides to check all characters exchanged during SpaceWire link initialization, as shown in Fig. 8. That is another good example of SpW DECODE usage.



Fig. 8. SpaceWire Link Initialization

After each side gets to Run state, transmission rate is switched from 10 to 200 Mbps.

CONCLUSION

This paper presented SpaceWire related features of a R&S RTO oscilloscope. The SpaceWire DECODE capabilities of the scope shows SpaceWire characters on the screen at same time that user could pay attention to issues like transition time and glitches. SpaceWire TRIGGER feature allows user to find a particular data or control character or a rare event (such a parity error for example).

Two methods to measure SpaceWire Data-Strobe SKEW were presented and compared, with the result that they are equivalent. EYE diagrams were extracted from SpaceWire Data and Strobe signals, which shows combined effect of skew and jitter. The effect of different cable lengths were investigated on SKEW measurements and also on EYE diagrams.

Cross correlation between SpaceWire messages (decoded), SpaceWire signals (analog) and power consumption was also shown as a powerful tool.

Taking all these different usage scenarios together, we can conclude that this scope is appropriated to debug issues related to the lower layers of SpaceWire Protocol Stack.

ACKNOWLEDGMENT

CTI people thanks CITAR Project, and support from FINEP, CNPq, and FAPESP. We also thank AXON for providing us with some SpW connectors, and R&S Brazil for the logistics related to scope and probes.

REFERENCES

[1] SpaceWire Standard Revision Draft F1 Issue 1.7
[2] R&S RTO Digital Oscilloscope User Manual –

RTO_UserManual_en_10.pdf - 1332.9725.02-10

Missions & Applications (Long & Short)

Imaging NOGAH Software-Defined System Comprising Many RC64 Processors for Optical and SAR Observation Satellites

Missions and Applications session

Ran Ginosar, Peleg Aviely, Tsvika Israeli, and Fredy Lange Ramon Chips Ltd. Yokneam, Israel ran@ramon-chips.com

Abstract— The NOGAH system, based on multiple 6U-220 VPX cards and multiple RC64 manycore processors, provides a complete end-to-end solution for software-defined digital earthobservation payloads. It can be adapted to either panchromatic and hyperspectral optical payloads or to advanced SAR systems. The cards of a typical high-performance payload are described. A SpaceFibre network serves the cards through the VPX backplane, providing high throughput among the cards for flexible operation modes.

Index Terms— SpaceWire, SpaceFibre, VPX, RC64, EOS, SAR, Software defined payload, NOGAH.

I. INTRODUCTION

NOGAH offers four principal EOS services: compression, storage, processing, and communications. Adaptive compression can be applied at very high data rate by parallelizing the traffic and processing. The extremely large solid-state mass memory is managed closely by RC64 processors for resilience and data integrity, and can store images before and after compression, to enable powerful missions providing unlimited imaging capabilities. Processing cards facilitate image analysis, real-time target recognition and identification, precise pixel registration, data fusion, and machine learning-based analysis. EOS also requires extensive communications capabilities to beam images to the ground directly or through other satellites, and to transfer images coming from other satellites. NOGAH systems include modems, switching, and routing to facilitate such high-end communications. Software can be uploaded and modified while in orbit.

NOGAH systems are based on the OpenVPX (VITA 46) standad. Backplane interconnect integrates multiple cards with



Fig. 1. Interface card

SpaceFibre links,



Fig. 2. RC64 ADC/DAC integration

providing high performance payloads for EOS. Camera interface cards may connect image sensors, transferring signals over SpaceFibre links to the backplane. SAR interface cards integrate multiple high-speed analog-to-digital (ADC) converters, transferring the data over SpaceFibre links to the backplane. Compute cards process incoming data towards storage or retrieve storage data for post-processing. Storage cards serve the compute cards for high volume data storage with high throughput. Communication interface cards provide downlink or inter-satellite links towards the ground station. Each card has internal routing and switching capabilities, supporting flexible routing of data among the cards, through the passive, mission-customized backplane.

II. NOGAH CARDS

A. Interface card

A typical interface card (Fig. 1) for RF input or output includes multiple ADCs/DACs with DDR LVDS interface to multiple RC64 processors, shown in Fig. 2. The clock generation network provides a synchronized clock source for the multiple ADCs/DACs and RC64s, which is good for phasealigned ADC/DAC arrays. The Leon3FT-based GR712RC controller is responsible for card management and the SpaceWire interface to the backplane. The backplane interface



Fig. 3. Compute card

includes 48 SpaceFibre links, at 6.25Gbps speed. Peak throughput of the DAC-based card reaches 144Gbps, for 12GSamples/sec. The ADC-based card provides similar input throughput. Combinations of the ADCs and DACs are supported as well.

B. Compute card

A typical compute card (Fig. 3) includes eight RC64 processors organized in clusters of four (Fig. 4), each connected to DDR3 SDRAM storage. Four of the processors connect to the backplane, each with eight SpaceFibre links, at 6.25Gbps speed. Peak throughput is 240Gbps input and 240Gbps output. The other four links connect to the RC64 cluster members. Management and clock generation are similar to the interface card.

C. Storage card

A typical storage card (Fig. 5) combines two clusters (Fig. 6) of RC64, rad-hard FPGA (e.g., Microsemi RTG4), and ten groups of 16 flash components. Three hundred and twenty high-end flash components (spread on both card sides), of storage SLC/MLC/TLC provide 80/160/240 Tbytes accordingly. The FPGA design includes ten ONFI controllers and bridge functionality with SpaceFibre to/from the RC64 processor. The peak throughput of 50Mbytes/sec x ten controllers x two clusters gives 1Gbyte/sec. The RC64 processor provides volatile memory buffering, forward error correction, and a file system layer to abstract the storage. Twenty SpaceFibre links connect to the backplane at 100Gbps input and output throughput.

III. NOGAH SYSTEM

The system in Fig. 7 includes an input card, two compute cards, a storage card, and an output communication card. The VPX backplane network topology in (A) provides the three main operation modes: (B) Capture, (C) Compute, and (D) Data transmit.



Fig. 4. RC64 compute cluster integration







Fig. 6. RC64 with flash integration

A. Capture (Fig. 7.B)

The interface card analog inputs can be synchronized using an input sync signal and a reference clock, provided by an external source. Each of the eight SpaceFibre quads can stream up to 1.5G samples/sec of a 12-bit/sample. RC64 handling of the input data includes packing data into packets and sending to the compute cards for processing.

The compute cards receive the data and perform processing according to the loaded program. Two RC64 processors handle a single data stream with up to 50GFlops and 100GMACs. Total performance can reach 400GFlops and 800GMACs. Results are sent to the storage card at a burst rate of 10Gbps per incoming data stream and a sustained data rate of 8Gbps.

B. Compute (Fig. 7.C)

Data from the storage card is loaded for processing at a throughput of 8Gbps and bursts of 80Gbps. Results are stored back, prepared for transmit stage.

C. Transmit (Fig. 7.D)

Data from storage is either sent directly to the output card or through compute card B. Processing may include radio signal generation for a wide-band down-link of up to 6GHz bandwidth.



Fig. 7. NOGAH system

IV. SUMMARY

The NOGAH system can handle high speed data capture, store the data, perform processing, and send the data to a ground station. Each card is programmable to perform various applications. Software upgrades in orbit provide mission flexibility and enable future modifications. System modularity enables building mission-specific configurations to suit target performance and redundancy. The SpaceFibre network is flexible, easily routed on the VPX backplane, and supports high throughput data streaming among the cards, with flow-control and FDIR capabilities. Leon3FT-based GR712RC controllers in each card control the system through the SpaceWire network.

SpaceWire as a Cube-Sat Instrument Interface

Missions and Applications – Long

Susan C. Clancy, Matthew D. Chase, Anusha Yarlagadda, Michael D. Starch, James P. Lux

Jet Propulsion Laboratory, California Institute of Technology

Pasadena, CA, USA

susan.c.clancy@jpl.nasa.gov

Abstract— SpaceWire is used in the control and data interface for an instrument on a pair of small satellites, one of which was launched in summer 2017. The instrument SpaceWire interface is implemented in a Field Programmable Gate Array as an instantiated core controlled by a LEON3FT CPU, which is also implemented as an instantiated core. The UT699 processor in the flight computer provides the spacecraft side's SpaceWire interface.

A simple message based protocol consisting of four message types was defined, based on existing SpaceWire standards. One was for passing commands to and responses from the instrument in the form of text strings similar to those from a system console where each line of text is passed in a SpaceWire message. Another was for passing spacecraft time to the instrument. The third was for transferring files using a subset of the Remote Memory Access Protocol (RMAP). The fourth was for retrieving science data from the instrument.

A set of user application programming interface (API) routines provided an abstracted interface to both the serial console (used during debug) and the SpaceWire device interface.

Early instrument development and testing was done with a set of utilities that controlled a Star-Dundee USB-SpaceWire brick providing a user interface similar to a serial console terminal emulator with the addition of file and data transfers. Later in the integration and test process, these utilities were integrated with the COSMOS ground systems software used for spacecraft control, providing a seamless transition from standalone instrument tests to benchtop flat-sat test and full spacecraft level tests.

Keywords—cubesat; SpaceWire; LEON3FT; RMAP; COSMOS; USB-Spacewire brick

I. INTRODUCTION

SpaceWire is used in the command and data interface between an instrument payload and the host spacecraft. The host spacecraft is a standard 3U (approximately 30cm x10cm x10cm) spacecraft from Space Dynamics Laboratory in Logan, UT. The spacecraft provides the support infrastructure: solar panels, batteries, flight computer, GPS receiver, and attitude control. The instrument is based on the JPL Iris radio [1] which uses a Xilinx Virtex 6 Field Programmable Gate Array (FPGA) that serves both as a digital signal processor and as the instrument controller. The instrument runs the RTEMS operating system [2] on an instantiated LEON3FT CPU core. SpaceWire is used to pass commands to the spacecraft as ordinary human readable strings, and the spacecraft sends back telemetry, also as human readable strings. The instrument is commanded to collect data for about 10 minutes at a user specified time. The digital signal processing algorithms produce data at about 5 Mbps which is stored in flash memory in the instrument. The ground operators command the spacecraft to requests the science data, which is streamed out to the spacecraft at high speed over the SpaceWire interface where it is stored in spacecraft flash memory. Ultimately, the spacecraft sends the data to a ground station when requested.

There are provisions for the spacecraft to transfer files to and from the instrument using a subset of the RMAP protocol [3]. In addition, the instrument receives periodic time updates passed from the onboard spacecraft GPS receiver once a second, so that the time of capture can be set accurately.

II. PHYSICAL IMPLEMENTATION

The SpaceWire interface on the instrument side was implemented using the Gaisler GRSpW2 core instantiated in the



Figure II-1 Spacecraft-Instrument Interfaces

Virtex 6 FPGA, along with the LEON3FT CPU core. The 4 LVDS pairs are directly connected to the pins with no external driver or receiver to minimize changes to the existing processor board design. The spacecraft Single Board Computer (SBC) [4,5] uses the UT699 CPU, which includes the SpaceWire interface as part of the chip. CubeSat's are physically small and compact, Omnetics NanoD (MIL-DTL-32139) connectors were used throughout the spacecraft, with twisted pair wires for the 5 cm cable. The SpaceWire interface used a 9 pin configuration, with the pin assignment identical to the usual SpaceWire Micro-D.

III. PROTOCOL

The interface between the instrument and the spacecraft uses four different variable length packet formats, each identified by a unique protocol identifier (PID), shown in Table III-1. The use of different PIDs allows easy separation of the messages when received, without needing to further parse the message to determine the content. For the most part, the spacecraft Flight Software (FSW) just passes the entire SpaceWire message through unchanged in either direction. The spacecraft command header is stripped off and the embedded SpaceWire command message is passed through to the instrument. The protocols described below are those processed by the instrument.

Table III-1 Protocol Ids

PID	PROTOCOL ID DESCRIPTION
0x01	RMAP – used for file and binary data transfer
	to/from the instrument
0xF0	text (ASCII) data to from the instrument (stdin,
	stdout)
0xF1	Sampled Data as VITA-49 packets returned from
	the instrument
0xF2	GPS Binary message to the instrument

The SpaceWire packets follow the format defined in the SpaceWire Specification [6]. The packet includes a destination address, payload of application specific data, and an End of Packet (EOP) marker as shown in Figure III-1.

For an instrument SpaceWire packet, the address was always 0xFE, regardless of whether it was sent to or received from the instrument since it was a point to point link. The instrument includes the Protocol Id (PID) which identifies the packet type, followed by the payload data, and ending with the Cyclic Redundancy Check (CRC) used to detect data corruption.

Address





A. Serial Console Emulation

The instrument software provides a serial console style interface with variable length text commands which produce variable length text response messages. These command and response messages are identified using the 0xF0 Protocol Id. The API routine used by the software to send text messages inserts a text sequence number and the instrument time before each message.





B. File Transfer To the Instrument

File transfers to and from the spacecraft use a subset of the existing RMAP protocol. These messages use the 0x01 Protocol Id and the RMAP format which includes the destination

Τι	A	Ox	0x01		uction	Key (0xA5)
Initiato	r Addr	Transactio	n ID (MSB)	Transactio	on ID (LSB)	z	ero
Address	s (MSB)	Address		Add	iress	Address (LSB)	
Length	(MSB)	Len	gth	Lengt	h (LSB)	Head	er CRC
Data	Data	Data	Data	Data	Data	Data	Data
Data	Data	Data	Data	Data	Data	CRC	EOP

Figure III-4 RMAP Write Format (data to instrument)

target logical address (TLA), RMAP header fields, data, RMAP CRC, and end of packet marker. The RMAP header fields identify the RMAP instruction, key, initiator address, transaction id, address, length as shown in Figure III.4. The address fields define the position within the file being transferred and the length is the length of the data portion within the message. Write Reply messages are returned to the spacecraft with the format given in Figure III-5 if a Write-With-Reply instruction is used.

Initiator Address	0x01	Instruction	Status
Instrument Addr	Transaction ID (MSB)	Transaction ID (LSB)	Header CRC
Figure III-	5 - Write Reply (fi	rom instrument, if	reauested)

To send a file, an FWRITE command is sent giving the name of the file and the maximum length of the file. The file length allows allocation of the space in the in-memory file system in advance of the transfer. Then all the RMAP Write packets are sent with the file contents. The payload data from each RMAP Write message is written to the specified file at the address in the message header. After all the RMAP data messages are sent, an FCLOSE command is sent. RMAP messages received when there is no active FWRITE are discarded with an error message.

C. File Transfer From the Instrument

Files are transferred from the instrument to the spacecraft by using RMAP Read messages as shown in Figure III-6. An FREAD command is sent to the instrument, and the instrument replies with a series of RMAP Read messages containing the file contents. The Transaction Id is used a s a sequence number and used to detect missing messages by the ground processing.

n	A	0x01		Instruction		Status	
Inst Logica	al Address	Transactio	n ID (MSB)	Transactio	on ID (LSB)	Ze	ero
Length	(MSB)	Ler	ngth	Lengt	h (LSB)	Head	er CRC
Length Data	(MSB) Data	Ler Data	Data	Lengti Data	h (LSB) Data	Head Data	er CRC Data

Figure III-6 - RMAP Read (data from instrument)

D. Science Data Transfer from the Instrument

The primary science output of the instrument is long streams of sampled data representing the received sensor signals. The data is encoded in the Virtual Radio Transport (VRT) packets defined in the VITA-49 specification [7]. A SpaceWire message was defined that contained the entire VRT packet with the 0xF1 protocol ID. A VRT packet stream consists of periodic context packets interspersed in a stream of data packets, each containing 250 samples. The context packet that precedes the data packets identifies information about the data stream.



Figure III-7 VITA-49 Data Packet

The command which starts sending science data allows a pacing delay to be inserted between each message, typically on the order of 10 milliseconds, to limit the overall data rate to the spacecraft.

E. GPS Time to the Instrument

The spacecraft uses a GPS receiver to determine the time, which is provided to the instrument via a GPS message sent as a 0xF2 Protocol Id message. The original design of the instrument used the time distribution message using a CCSDS Unsegmented Time proposed by Habinc, *et al.* [8]. However, to minimize the changes in the existing spacecraft flight software, a simpler message that encoded the GPS week and milliseconds was used, as shown in Figure III-8.



Figure III-8 GPS Packet Format

The GPS milliseconds represents the number of milliseconds that have elapsed between the GPS epoch and the next GPS onepulse-per-second tick (1PPS), which is received on a discrete input line. As each GPS message is processed, the given GPS time is saved and the system clock ticks are stored with the associated 1PPS tick. The instrument internal clock is latched on each 1pps, along with the last received GPS Milliseconds value, which is used to calculate GPS time from instrument internal clock time. Software logic detects missing 1pps pulses or GPS time messages that are out of sequence.

IV. SPACEWIRE API

The instrument software implements a simple message passing style API to provide a consistent interface to the SpaceWire hardware. Transmit and receive tasks handle data sent or received and the GAISLER SPW2 SpaceWire device driver library functions perform the low-level device operations with the hardware. The "spacewire_init" function combines some of the low-level function calls needed to initialize the hardware into a higher level API function call.

The API functions (Table IV-1) format data sent from the instrument into a SpaceWire message and queue it as a transmit request to the transmit queue (TxQ). An instrument user application sending data calls one of the three API "send" functions to send data as text (PID=F0), RMAP data (PID=01), or VITA-49 VRT data (PID=F1). The transmit task (See Figure

Table III-1 SpaceWire API

FUNCTION NAME	DESCRIPTION
spacewire_init()	Initialize and sets up the
	buffer for sending and
	receiving data
<pre>send_data_packet(len,tid,buf)</pre>	Send an RMAP data packet
send_text_packet(len,buf)	Send a text packet
send_vita49_packet(len,buf)	Send VITA-49 packet
send_write_reply_packet(len,	Send fwrite reply
id,buf)	
<pre>set_fwrite_params(fn, fsize)</pre>	Updates file IO name and
	size from the FWRITE
	command
write_packet_to_file(pkt)	Decodes an RMAP packet
	and writes the data to a file
dump_packet(buf,len)	Outputs packet data as a
	series of text messages with
	HEX ASCII encoded data
	values
print_diag_packet(buf,	Decodes and outputs the
len,opt)	contents of a packet in a
	series of text messages

IV-1) dequeues each TxQ entry and calls the SPW2 spw_tx() and spw_checktx() device driver functions. The spw_tx() function transmits the packet out over SpaceWire. The spw_checktx() function blocks until the packet is transmitted or fails with an error. The transmit success and fail statistics are updated and can be captured and reported by the "spacewire_status" command.



Figure IV-2 SpaceWire Transmit Software Architecture

The receive task (see Figure IV-2) handles three types of incoming packets which are identified by their PID. These are text commands (PID=0xF0), RMAP WRITE data (PID=0x01), or GPS data (PID=0xF2). The SPW2 spw_rx() and spw_check_rx() device driver functions are called within the receiver task to receive packets. The spw_rx() function sets up the input buffer to receive data and the check_rx() blocks until data arrives or detects a failure. The receive success and fail

statistics are updated and can be captured and reported by the "spacewire_status" command.



Figure IV-2 SpaceWire Receive Software Architecture

Incoming text messages are forwarded to the command input message queue which is serviced by the command processor. Any incoming RMAP WRITE data packets are decoded and the data is written to the file previously opened by an FWRITE command. Any incoming GPS data packets are processed by calling the GPS API.

The integrity of the incoming SpaceWire messages is checked using the data CRC and, if it matches the expected CRC value, the message is handled. If not, a receive error count is incremented and the message is discarded.

There are 4 API entry points shown in Table IV-1 which map to specific commands that the instrument can receive: fwrite() and fclose() used before and after importing data into a file, a "configure" command to enable diagnostic messages, and a "status" command to report SpaceWire API statistics.

Table IV-1 SpaceWire Related Commands API

FUNCTION	DESCRIPTION
fclose_cmd	Closes a file previously opened
	by the FWRITE command
fwrite_cmd	Creates a zero-filled file of the
	specified length in preparation
	for writing incoming RMAP
	data
spacewire_cfg_cmd	Configures SpaceWire API
	diagnostic option on or off;
	When diagnostics are on, the
	contents of sent and received
	messages is output for
	debugging
spacewire_status_cmd	Reports SpaceWire API
	statistics

V. GROUND SUPPORT SOFTWARE

The spacecraft ground system and much of the testing uses the open source COSMOS system from Ball Aerospace [9, 10]. The COSMOS system uses the Ruby language with procedures and modules written in Ruby to run test sequences, send commands, receive and format telemetry. During the development of the instrument, we had 3 fundamentally different use configurations:

- Test Bench Configuration Standalone instrument on the bench with a Star-Dundee USB-SpaceWire brick
- FlatSat Configuration Instrument connected to flatsat, using a RS422 link
- Spacecraft Configuration Instrument integrated into the spacecraft

The Test Bench Configuration had a laptop computer with the Star-Dundee USB SpaceWire brick and associated tools and drivers connected via SpaceWire to the instrument. This configuration was used for the initial development and testing of the instrument hardware and software.

The FlatSat Configuration is a breadboard configuration including non-flight qualified instances of the spacecraft Single Board Computer and Interface Card. The spacecraft radio is emulated in the FlatSat by a serial port connected to a PC. The FlatSat includes a flight-like SpaceWire interface with an uplink / downlink radio interface simulated over an RS-232 connection.

The Spacecraft Configuration has the instrument integrated into the spacecraft and uses a ground station RF link for the uplink/downlink interface.

These configurations were accommodated by creating a Ruby software class hierarchy (see Figure V-1) that supported a standard (configuration independent) interface for interacting with the instrument. In this way, all command scripts, unit-tests, and other ground utilities operate with the instrument without



Figure V -1 Different configurations support a standard interface

needing to be altered when the physical configuration changed. Simply using a class's method, *e.g.* "send_inst_command" allows the tests to works on any interface or configuration. Ground processing software only needed to be written once to support all three testing configurations.

Ruby classes supporting the generic interface were created to automatically separate the incoming telemetry, a combination of instrument ASCII, file data, and VRT packets, into appropriate streams. These streams supported testing of the instrument and software in a flight-like environment.

ACKNOWLEDGMENT

This work was carried out at the Jet Propulsion Laboratory in Pasadena (JPL), California, under contract with the National Aeronautics and Space Administration. References herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not constitute or imply its endorsement by the U.S. Government or the Jet Propulsion Laboratory.

©2018 California Institute of Technology. Government sponsorship acknowledged.

REFERENCES

- Jet Propulsion Laboratory, "Iris V2.1 CubeSat Deep Space Transponder", <u>https://www.jpl.nasa.gov/cubesat/pdf/Brochure_IrisV2.1_201611-URS_Approved_CL16-5469.pdf</u>
- [2] OAR Corporation, Real Time Executive for Multiprocessor Systems RTEMS, https://www.rtems.org/
- [3] European Cooperation for Space Standardization, "Space Engineering; SpaceWire Links, nodes, routers and networks,", ECSS-E-ST-50-12C, July 2008
- [4] Space Dynamics Laboratory, "PEARL Spacecraft Platform", <u>http://www.sdl.usu.edu/downloads/pearl.pdf</u>, downloaded 25 Feb 2018.
- [5] Q.Young, R. Burt, M. Watson, L. Zollinger "PEARL CubeSat Bus-Building Toward Operational Missions", Cubesat Summer Workshop -8-9 August 2009, Cal Poly San Luis
- [6] European Cooperation for Space Standardization, "Space Engineering; SpaceWire Links, nodes, routers and networks,", ECSS-E-ST-50-11F, Dec 2006
- [7] ANSI, "ANSI/VITA 49.0-2015 VITA Radio Transport Standard", <u>https://shop.vita.com/ANSI-VITA-490-2015-VITA-Radio-Transport-VRT-Standard-AV490.htm</u>
- [8] S. Habinc, A. Sakthivel, M. Suess, "SpaceWire Time Distribution Protocol", 2013 International Spacewire Conference.
- [9] R. Melton, "Ball Aerospace COSMOS open source command and control system," Small Satellite Conference, August 2016.
- [10] Ball Aerospace, "COSMOS The user interface for command and control of embedded systems," <u>http://cosmosrb.com/</u>, retrieved 25 Feb.

Spacehand: a multi-fingered robotic hand for the use in space

SEDLMAYR HANS-JUERGEN*, Bayer Ralph*, Bertleff Wieland*, Beyer Alexander*, Chalon Maxime*, Friedl Werner*, Maier Maximilian*, Obermeier Thomas* and Willberg Bertram*

> *Institute of Robotics and Mechatronics (RM) German Aerospace Center (DLR) Muenchener Str. 20, 82234 Wessling, Germany Phone: +49(0)8153-28-3532 Fax: +49(0)8153-28-1134 Hans-Juergen.Sedlmayr@dlr.de

Abstract—Robotic systems are not yet ready to replace human presence in space, but they provide an essential support for astronauts during routine tasks. Combined with telepresence technologies, they are ideal for hazardous or remote tasks, enabling new applications while reducing the risk and the costs associated with space operations. The Spacehand, a multi-fingered robotic hand is based on the former DEXHAND which was a technology demonstrator for a space qualifiable robotic hand. Unlike the DEXHAND, which was designed for a short term mission in ISS environment, the Spacehand shall survive a multi year servicing mission on the geostationary orbit. During the so called "Robotic-Servicing of Geosynchronous Satellites" - mission (RSGS) of the Defense Advanced Research Projects Agency (DARPA), the Naval Research Laboratory (NRL) and the Space Systems Loral (SSL), the Spacehand shall be utilized. To meet the requirements of the RSGS mission, the electronics and the mechanics of the hand must be adapted as well as the communication interface which was changed from CAN bus to SpaceWire. On top of the SpaceWire bus a Protocol related to the "Reliable Data Transfer Protocol" is realized. Furthermore the DSP code can be updated during the mission by means of a special SpaceWire bootloader.

Index Terms—Space, Robotic, multi-fingered robotic Hand, Spacewire, Spacehand, orbital servicing

I. INTRODUCTION

Robotic systems are used to explore the solar system, today. Several systems are landed on distant planets and asteroids and delivered lots of data and fascinating pictures. In near future, they shall be able to support humans during their risky work in space. Thanks to tele-presence technologies they are ideal for hazardous or remote tasks. These enable new applications like on-orbit-servicing, assembly and repair while the risk for humans and the costs of the missions decreases. Therefore, the development of more dexterous and more compliant technology is included in the road map of multiple space agencies and space companies.

The DEXHAND and DEXARM [1], [2] projects have been done in collaboration with the European Space Agency (ESA) and investigated the potential of hand and arm combination for support of Extra Vehicular Activities (EVA). The final acceptance test of the DEX-HAND (see Fig. 1) demonstrated the high capabilities of a multi-fingered robotic hand for common space tasks. Similar projects are led by JPL, with the hand of the Robonaut [3], or at the University of Laval, with the under-actuated SARAH hand [4]. After the completion



Figure 1. Dexhand with a pinch grasp

of the DEXHAND project, several space agencies have manifested their interest to include the hand in a real mission and only 2 years after the end of the DEXHAND project, the Spacehand project started. It is a contribution to the "robotic-servicing of geosynchronous satellites" mission (RSGS) of DARPA, NRL and SSL. During this mission, the Spacehand shall be used as a universal tool among other more specialized ones in a multi-year, geosynchronous orbit mission [5].

However, while the DEXHAND was designed for a short term low earth orbit (LEO) operation environment, the Spacehand must be adapted to meet the requirements for the RSGS mission. This paper reports the design of the Spacewire communication bus and the development of the SHRDDP, a customized version of the RDDP protocol for the Spacehand. The work reported is performed at the Institute of Robotics and Mechatronics under internal funding.

II. SYSTEM OVERVIEW

The design of the Spacehand is based on the long standing history of developments of torque controlled multi-fingered hands [6]–[9] at the institute. The team has a great knowledge database in the design and control of multi-fingered robotic hands and their integration in large mechatronic systems. The experience gathered during the development and operation of the DEXHAND proved to be extremely valuable in improving the safety margins of the design.

A. Hand structure

The Spacehand has four fingers with four degrees of freedom each, with the last two being coupled, resulting in a total of twelve actuated DOFs. Fig. 2 presents one DEXHAND finger. The design of the Spacehand is based on similar modular fingers and motor modules since it allows to produce a large number of identical and, consequently, high quality units. The joints in the fingers are actuated by tendons, mechanically powered by the motor modules. These motor modules fit into a cylinder of 27 mm diameter and a length of 20mm with a weight of 50g and provide a continuous torque of 2.4 Nm with peaks up to 9 Nm which is the maximum peak torque of the gearing.

B. Electronics

The electronics of the DEXHAND was designed to meet the requirements of a short term LEO mission. Therefore, the electronics of the Spacehand was modified to meet the requirements of a multiple year GEO mission. The experience gathered with the DEXHAND guided a number of optimizations for the new electronics.



Figure 2. Finger without housing

Furthermore the communication bus was modified from CAN to Spacewire.

One major design driver was the increased radiation requirement, which led to a new motor controllers design. The selected, automotive rated, brushless DC motor controller was tested in house. Up to 550 Gy(Si) all measured parameters stayed within their rated limits. Fig. 3 shows the TID response of the motor controller over dose. An heavy ion test campaign was conducted as well. No severe effect could be measured up till a LET of 59.9 MeV.cm2/mg.

Finally, the software running on a DSP/FPGA imple-



Figure 3. Digital supply current (12V) over dose

ments the communication mechanisms, the impedance control algorithm and several management functions.

III. SPACEWIRE

The electrical and low level protocol of the Spacewire communication interface is defined in the Spacewire standards ECSS-50-11 and ECSS-50-12C. The Spacehand is using a Spacewire core provided by StarDundee for this low level implementation. A FIFO based binding to the core allows the DSP to access the receive and send buffer, as well as the status and control registers through a dual-ported RAM.

However, the Spacewire standards are only specifying a relatively primitive communication interface, offering an extremely flexible and bandwidth efficient communication channel, delegating the protocol design to the user. In its native form, only the link establishment rules, the routing and some special operations are defined. The packets have a target ID but not a source ID, making it challenging to implement synchronous communication. The packets are not limited in size, although most known implementations set a limit for implementation reasons. The packets are not uniquely identified, creating potential security risks, but more important, issues during the reconstruction of segmented payloads.

For the mission, it may appear that the communication has the lowest quality; that is, packets might arrive out of order, packet might get corrupt, packet might be dropped, packets might be buffered and the communication has a delay up to 3 seconds. The goals of the protocol are to ensure a reliable communication, maintain an acceptable throughput, use a fix amount of memory on the Spacehand and provide a upper bound on the computation effort required to process the packets. In the case of Spacehand, there are several types of communications:

- The realtime streaming of telemetry data, of fixed size and fixed rate.
- The realtime, synchronous commands of the ground control.
- The realtime, synchronous telemetry of the Spacehand, warning and error messages.
- The non-realtime, synchronous upload of a software image (patches)
- The non-realtime, synchronous download of high speed recordings stored in the Spacehand.

The use cases are very similar to the ones of ethernet which unfortunately, due to its numerous headers has a fairly low payload efficiency, in particular for small payloads. Considering that the routing is achieved by the low level link, a direct implementation of UDP/TCP over spacewire would be extremely inefficient. The GRDDP (GOES-R Reliable Data Delivery Protocol) [10], proposes a design that satisfies most of the requirements but contains redundancies and lacks data that is required to deal with long delivery delays in an efficient manner. Nonetheless, many elements of the GRDDP have contributed to the design of the Spacehand protocol, named SHRDDP (Spacehand Reliable Data Delivery Protocol). It is important to note that although the GRDDP format might already be used by several systems, the relative rarity of such Spacewire systems means that providing GRDDP compliant software is not a major issue and, as the protocols are similar, can be adapted if the mission requires it.

A. SHRDDP Structure

The SHRDDP packets are constructed as depicted in 4. The fields are described in Table I



Figure 4. SHRDDP Packet Structure

Table I SHRDDP PACKET FIELDS

Field	C-Type	Description
DEST ID	uint8 t	ID of the destination node
SRC ID	uint8_t	ID of the source node
STATUS	uint8_t	Status byte of the Spacehand
TV IDU	uinto_t	ID of the last transmitted mediat
	unitio_t	ID of the last transmitted packet
TX_IDL		
RX_IDH	uint16_t	ID of the last received packet
RX_IDL		
MSCNT_H	uint8_t	local millisecond counter
MSCNT_L		
OPTYPE	uint8_t	type of operation,
		used to depack OPDATA
OPLEN	uint8_t	length of the OPDATA field
OPDATA	uint8_t[]	data of the operation
CRC8	uint8 t	checksum of the packet payload

B. SHRDDP Field Description

In the current scenario, the IDs are

- 0xD0 SHRDDP_GROUND_STATION_ID
- 0xC0 SHRDDP_SPACEHAND_ID
- 0xB0 SHRDDP_SATELLITE_ID

The STATUS byte is a very important field. It is used to signal emergency states through the system without the

need to interpret the message payload. It is described in detail in Table.II and Table.II

Table II SHRDDP STATUS FIELD BITS SPACEHAND

7	6	5	4	3	2	1	0
EF	TF			mode[3]	mode[2]	mode[1]	mode[0]

 Table III

 SHRDDP STATUS FIELD BITS DESCRIPTION SPACEHAND

EF	Error flag, Set if the hand requires the arm to stop
TF	Transmit flag. Set if the hand requires
	the message to be forwarded to ground
Mode	Current operation mode of the hand

In particular, it allows the arm to stop any motion without any action of the ground station (which might see the failure only after several seconds). Conversely, the arm or the carrier satellite can require an emergency stop of the Spacehand, once again, without requiring a ground command. The field also serves as a secondary routing mechanism, signaling if the message shall be forwarded to ground or not. Indeed, the amount of data resulting from high definition, long time recording of an operation, is too large for the limited on board storage of the Spacehand. Therefore, the telemetry can be directed toward the satellite computing infrastructure, logged and transmitted to ground in bulk, at a later time, for analysis. Nonetheless, some of the telemetry packet shall be forwarded to ground to allow the operator to monitor the state of the hand as well as all emergency and synchronous communications. Finally, the lowest bits of the STATUS byte are used to report the operation mode of the hand and simplify the monitoring of the arm/hand operations. In the event of an invalid CRC8 value, the packet shall be considered invalid and the EF and TF flags shall be considered set. The STATUS field has a different meaning for incoming messages. The bits position and description are reported in Table IV and Table V

 Table IV

 SHRDDP STATUS FIELD BITS GROUND STATION

7	6	5	4	3	2	1	0
EF	ATX						

The MSCNT field provides a local time tracking that allows to reconstruct the telemetry exact sampling points. It is important to note that the value wraps

 Table V

 SHRDDP STATUS FIELD BITS DESCRIPTION GROUND STATION

- EF Error flag, set if the ground station requires the hand to stop
- ATX Accept all TX. Set if the ground station requires the hand to accept the packet even if the TX_ID is not correct

around approximately every minute. The current version of SHRDDP does not synchronize the counter with the ground. The TX_ID, RX_ID fields are used to implement the reliability of the transmission. TX_ID and RX_ID are the identifiers of the last packet transmitted and the last packet accepted (that is, the last valid packet) by the node. They are both uint16_t values which wrap around at the upper limit. The OPLEN, OPDATA and the OPTYPE fields are used to describe the payload and interpret it. The OPLEN field is the number of transported OPDATA bytes. It shall match the number of bytes excepted by the OPTYPE byte. It does not include the header length nor the CRC8 byte. The minimum value is 0 and the maximum is 114. (128 maximum packet size offered by the FPGA implementation, minus the logical address and EOP byte, minus the SHRDDP header and the CRC8 byte). Packets with invalid OPLEN values shall be considered as invalid. Data is always transmitted in integer types with 2's complement and without padding. Both nodes are responsible of packing and unpacking the data. Care must be taken to deal with non-aligned accesses and padding can be inserted to simplify the implementation.

C. SHRDDP Reliability Concept

Implementing reliability over a non-deterministic, nonreliable medium results in a trade-off between memory resources and bandwidth usage efficiency. In the nominal case, described in Fig. 5, packets are sent in a strobe from one node, e.g. the Spacehand, received by the other node, e.g. the ground station, then the process is reversed. In case of emergency, for example when the ground station detected an anomaly, the ground station sends error packets with the ATX flag set such that the packet is accepted even if it is received out of order. This case is depicted in Fig. 6.

In the case of a packet loss, either due to a transmission error or a link drop, the receiver will receive packets out of order and will not accept it. This case is depicted in Fig. 7. It will keep sending its own packet with the last received identifier until the sender node receives it and retransmit the queue from this last identifier. In









EMERGENCY COMMAND

Figure 6. SHRDDP Reliability Emergency Command

order to avoid a dead-lock situation, it is necessary that the receiver keeps announcing its last received identifier, either through a special acknowledge message or simply through its regular telemetry stream. Thanks to this mechanism, each node is ensured to process the packets in order, at the cost of increased communication delay. Each node is free to use a windowing mechanism in order to improve the communication bandwidth, indeed, in the nominal case, all packets will arrive in the order and will not be dropped by the link.



TRANSMISSION LOSS

Figure 7. SHRDDP Reliability Transmission Loss

IV. CONCLUSION

This paper presented an overview of the Spacehand project and highlighted the Spacehand Reliable Data Delivery Protocol. This communication is based on a Spacewire LVDS connection with a link rate of 10 MHz. This speed is needed for the mission but its design should be able to be operated at higher communication speeds according to the Spacewire standard.

Acknowledgment

The authors would like to thank the Spacehand team at DLR-RM as well as the DARPA, NRL and SSL for the collaboration opportunity. Thanks are also addressed to the "Government of Upper Bavaria / Regierung von Oberbayern" for the funding support.

REFERENCES

- [1] A. Wedler, M. Chalon, and al., "DLRs space qualifiable multifingered DEXHAND," *ASTRA*, 2011.
- [2] A. Rusconi, P. Magnani, T. Grasso, G. Rossi, J. G. Lodoso, and G. Magnani, "DEXARM- a dexterous robot arm for space applications," *ASTRA*, 2004.
- [3] M. A. Diftler, R. O. Ambrose, S. M. Goza, K. Tyree, and E. Huber, "Robonaut Mobile Autonomy: Initial Experiments," *Robotic and Automation, IEEE International Conference on*, pp. 1437 – 1442, 2005.
- [4] B. Rubinger, M. Brousseau, J. Lymer, C. M. Gosselin, T. Laliberté, and J.-C. Piedboeuf, "A novel robotic hand - SARAH for operations on the international space station," *Proceeding* of the ASTRA 2002 Workshop, 2002.
- [5] DARPAtv, "DARPA Phoenix Concept Video," https://www. youtube.com/watch?v=OeKzdk0sWjI, accessed: 2015-04-29.
- [6] J. Butterfaß, G. Hirzinger, S. Knoch, and H. Liu, "DLR's multisensory hand part I: Hard- and software architecture," *Robotic and Automation, IEEE International Conference on*, 1998.
- [7] C. Borst, M. Fischer, S. Haidacher, H. Liu, and G. Hirzinger, "DLR hand II: experiments and experiences with an anthropomorphic hand," in *Robotic and Automation, IEEE International Conference on*, 2003, pp. 702–707.
- [8] Z. Chen, N. Y.Lii, T. Wimboeck, S. Fan, M. Jin, C. H. Borst, and H. Liu, "Experimental study on impedance control for the fivefingered dexterous robot hand DLR-HIT II," *Intelligent Robots* and Systems, IEEE International Conference on, 2010.
- [9] M. Grebenstein and P. van der Smagt, "Antagonism for a highly anthropomorphic hand arm system," *Advanced Robotics*, no. 22, pp. 39–55, 2008.
- [10] N. G. S. F. Center, "Geostationary operational environmental satellite (goes), goes-r series, goes-r reliable data delivery protocol (grddp)," NASA, Tech. Rep., 2005.

Components 1 (Long)
The Evolution of SpaceWire Electrical Interconnect

Faster, Lighter, Smaller

Nigel Kellett Axon' Cable LTD Dunfermline, UK <u>n.kellett@axon-cable.co.uk</u>

1 Introduction

As a manufacturer of classic Micro-D connectors and also of a wide variety of cables and interconnect used in Space, Axon' Cable is very well versed in the pros and cons of "classic" 9-way micro-D SpaceWire links. In previous SpaceWire conference papers Axon' presented its developments in Low Mass SpaceWire cable, and also an overview of the new MicroMach[®] impedance-matched SpaceWire connector development.

In this paper, Axon' reviews key results from the ongoing evaluation testing of this new MicroMach[®] SpW connector, developed under ESA Technology Research Project TRP AO/1-7985. In addition, a performance comparison is provided between the existing and the new technologies, allowing users to make an informed choice according to their differing system requirements.

For additional clarification, the paper also provides a brief recap of all Axon's recent developments in physical layer componentry, and where they sit with respect to the various ESA standards and specifications, either published or in progress.

Finally, Axon' explores a novel new concept of potential SpaceWire transmission over flat flexible cable (FFC), and the paper looks at some other potential wider uses for the MicroMach[®] connector.

2 Variants Developed

For the Evaluation phase, 3 different PCB variant connectors have been developed and manufactured.

2.1 MicroMach panel-mount SMT Variant

This version is used to interface to a PCB while optimizing space (edge PCB connection)



Fig-2.1-1 MicroMach SMT Variant exploded view

Stéphane Hermant Axon' Cable SAS Montmirail, France <u>s.hermant@axon-cable.com</u>



Fig 2.1-2 View of the soldered termination on the PCB

2.2 MicroMach panel-mount Wired Variant

This version allows greatest flexibility to make connections to the PCB with flexible twisted pair outputs.







Fig 2.2-2 View of the soldered termination on the PCB

2.3 MicroMach panel-mount Flex Variant

This version helps create a rapid and safe connection to the PCB using a matched impedance flex circuit. Terminations are compatible with castellated chip carriers, devices which can be soldered in one step.



Fig 2.3-1 MicroMach panel mount Flex Variant exploded view



Fig 2.3-2 View of the Flex soldered termination on the PCB

2.4 Inline Connector Variants

Concerning the inline connectors, AXON' has developed two principal variants: the male and the female panel-mount:



Fig 2.4-1 Male inline connector exploded view



Fig 2.4-2 View of a male cable termination







Fig 2.4-4 View of a female cable termination

3 Improved mating with mechanical guidance

To secure the mating sequence, two special guide pins are used which, as well as securing the backshell to the connector, makes "blind mating" possible.



Fig 3-1 Guiding sequence

4 Performance Comparisons / maximum speed

From the electrical Evaluation Testing carried out to date, we can see the following key differences between SpaceWire links made using the classic 9pin micro-D, and the new MicroMach[®] adapted

connector. The media is also a possible way of improvement using parallel pair instead of twisted one.

4.1 Crosstalk comparison

In comparison to the standard 9way Micro-D cabling, the MicroMach[®] variant offers a significant improvement in crosstalk (around -20dB less @ 1Ghz). Thanks to the 4 independent 100 Ohm cavities, the signal coupling between contacts pairs is reduced to a minimum.



Fig 4.1-1 Worst case Crosstalk MD9 versus MicroMach® link

4.2 Eye Pattern comparison

One of the requested improvement features achieved within the study was to have a 100 Ohms matched characteristic impedance between each pair of signal pins, which gives rise to improved signal integrity as can be seen in the screenshots below: The overshoot on the signal measured on the Micro-D terminated link is much more significant than with the MicroMach[®] version.



Fig 4.2-1 Eye Pattern on Micro-D 9pins terminated 1m link @400Mb/s



Fig 4.2-2 Eye Pattern on MicroMach® terminated 1m link @400Mb/s

4.3 Shielding effectiveness

With enhanced EMC design, the MicroMach[®] variant offers a noticeable improvement in shielding effectiveness (around -10dB better up to 18Ghz – the maximum frequency tested). This improvement is possible, partly thanks to the improved inner and outer shield termination between cable and connector, and partly due to improved EMC interface between male and female connector bodies.



Fig 4.3-1 Shielding effectiveness measurement on test vehicles. One is terminated with standard 9pins Micro-D connectors, the other with MicroMach[®].

4.4 Maximum speed

A cable with four, impedance-matched twisted shielded pairs is the baseline for SpaceWire links, however, in an effort to try and further improve electrical features, Axon' added a cable variant with four parallel pairs as two of the test vehicles in the study. This cable improves two of the key electrical features : Skew & Insertion Loss.

The maximum speed reached on a 4m link was performed on this new variant cable made with parallel pair compared to the qualified ESCC3902 version with twisted pair construction. The maximum $_{30}^{20}$

speed on this 4m test vehicle is located between 800Mb/s and 1600Mb/s as shown in the screenshots below.



Fig 4.3-1 Eye Pattern on MicroMach terminated 4m parallel pair link @800Mb/s



Fig 4.3-2 Eye Pattern on MicroMach terminated 4m parallel pair link @1600Mb/s

5 Recap of Axon' high data rate componentry

For additional clarification, the chart in Fig 5-1 provides a brief recap of all Axon's recent developments in physical layer componentry, and where they sit with respect to the various ESA standards and specifications, either published or in progress.

Component	Standardised?	Features and Advantages
"Classic" SpW cable, AWG 26		
and AWG 28	ESCC3902 002 and 003	Long heritage. Many programmes
Space grade MDSA micro-D		Long heritage, existing Type A SpW connector
connector	EPPL2 iaw ESCC3401 029	Potted, non-dismountable, robust connector
		Good early heritage developing (Solar Orbiter, JUICE,
		ExoMars, Metop)
		Half the mass, easier screening arrangement, more
Low Mass SpW cable, AWG 28	ESCC3902 004	flexible, lower bend radius, better radiation resistance
		Adapted 100 ohm impedance, better EMC design, much
MicroMach [®] adapted SpW		improved crosstalk, improved skew
connector	Not yet, TRP in progress	Suitable for both SpaceWire and EtherSpace applications
		Very high data rate (10GBps per channel) connector,
		available in single pair to 4 pair versions. Suitable for
AxoMach® HDR connector	Pending ESCC3907 001	SpaceFibre and WizardLink applications
		As above, in cable assembly form, using a pair of RF
		quality Ax2.4 coaxial cables per way
		Heritage on all AxoMach® product includes MAVEN,
AxoMach [®] HDR assemblies	Pending ESCC3907 002	NISAR, MTG, SENTINEL, SaRAH and SWOT
		A variant of the HDR connector range, optimised for single
		channel SpaceFibre transmission, with all 4 coaxial
AxoMach® SpFi connector	Pending ESCC3907 001	contacts space optimised within the connector body.

Fig 5-1 Chart of Axon' HDR components and their status

Additionally, the chart in Fig 5-2 provides a pictorial representation of some of the key products mentioned, and where they sit in terms of intended usage, either by increasing data rate, or by the type of media or network protocol.



Fig 5-2 HDR components organized by data rate and application

6 Faster, Lighter, ... Flatter ?

Aside from Axon's well-known activities in space, micro-D manufacture and high data rate products, the company is also a large manufacturer of Flexible Flat Cable (FFC), commonly used in consumer electronics, office automation and automotive applications. Axon' Flexible flat cable is used in around 20% of the world's cars, providing a clock-spring-style connection behind the steering wheel to drivers' airbags.



Fig 6-1 A typical FFC harness for automotive air-bags

A first prototype of a **Flat Flexible SpaceWire link** has been assembled using shielded flat cable and a new, flat shaped connector derived from the MicroMach[®] design parts with electric welding technology. This trial product opens the possibility of a new automated manufacturing solution improving flexibility and saving space. At the same time the media is improved in skew and characteristic impedance regularity features.



Fig 6-2 Flat shaped MicroMach® connector termination



Fig 6-3 Overview of the FFC MicroMach® SpW prototype

The benefits of such a product are principally in space and weight saving. The cable, even with shielding, is less than 1 mm thick, and can therefore run in and around tight areas previously impossible with conventional round cable, and additionally can be bent within a very small radius.

Initial testing of this first prototype has shown quite promising performance, even up to 400 Mb/s, and over a reasonable sample length of 2m.



Fig 6-4 Eye Pattern on 2m FFC SpW prototype link @400Mb/s

Reducing the bit rate a little to 300Mb/s, and extending the length right up to 10m still manages to achieve a (borderline) satisfactory performance.



Fig 6-5 Eye Pattern on 10m FFC SpW prototype link @300Mb/s (measured skew around 1ps/m)

Shielding designs are ongoing to select the best compromise between shielding efficiency and size/flexibility. Shielding Effectiveness simulations have been performed with different shield configurations e.g. one side shielding and 360° shielding.



Fig 6-6 "double sided ground plane" shielding SE simulation



Fig 6-7 Full 360° wrapped shielding SE simulation

7 Further possibilities for MicroMach®

7.1 Possible use of the new MicroMach connector for Ethernet, XAUI ("Zowie") or EtherSpace applications :

There are other possible alternative uses for this connector, due to its electrical features and pin allocation. Ethernet cable can be mounted on it, in order to build a robust Ethernet link. The Ethernet cable construction is close to that of SpaceWire and can be an interesting solution to interconnect equipments up to 10Gb/s. AWG28, 27 and 26 are all compatible with the MicroMach[®] termination. The possibility to use AWG24 is not yet developed but is a potential future development.

7.2 Possible use of the new MicroMach[®] connector for SpaceFibre applications.

A MicroMach[®] connector variant will be compatible with Axon' space approved 2.4mm coaxial cable to propose an alternative solution for very high data rate links such as SpaceFibre or to increase the length for common mode links (e.g CML technology extensively used in ICs).

In overall summary, we can see that the MicroMach[®] system has an exciting future. It electrically out-performs the 9pin micro-D in nearly all aspects: crosstalk, impedance matching, skew, shielding effectiveness, and EMC design for both inner and outer shield termination and mating. It is a versatile solution and can be used with classic SpW cable in both AWG 26 and 28, as well as Low Mass SpaceWire. It also has promising potential use as a higher data rate option for multi-gigabit protocols such as SpaceFibre or EtherSpace.

The traditional micro-D connector cannot be discounted just yet, however. It was initially selected for a combination of its small size and robust design, and it remains very effective for both these reasons. As data rates increase the 9pin micro-D is likely to eventually become obsolete in HDR protocols, but this tiny workhorse is set to be with us for still some time to come.

Keywords – Ethernet, EtherSpace, High Data Rate, HDR, Low Mass, Micro-D, MicroMach[®], Flexible Flat Cable, FFC, SpaceFibre, SpFi, SpaceWire, SpW, XAUI, Zowie



Fig 7-1 MicroMach[®] link with Low Mass SpW cable at SpaceTech Expo, Bremen, with STAR-Dundee SpW test equipment

Status Update on New Standard Products with SpaceWire Support

Update on GR740, GR718B and GR716 Standard Products

Components, Long paper

Nils Johan Wessman, Fredrik Johansson, Francisco Hernandez, Fredrik Sturesson, Jan Andersson Cobham Gaisler AB Gothenburg, Sweden

Abstract— The GR740 Quad-Core LEON4FT microprocessor, the GR716 LEON3FT microcontroller and the GR718B SpaceWire routing switch are three standard products from Cobham Gaisler that have SpaceWire interfaces as a key feature.

Index Terms— SpaceWire; LEON4FT, LEON3FT, microprocessor, microcontroller, routing switch, router, system-on-chip, rad-tolerant

I. INTRODUCTION

Cobham Gaisler provides three main standard products that implement SpaceWire as a key feature. The three devices are at different stages in product life cycle. The GR718B SpaceWire router [1] flight models lot qualification has been completed. New silicon for the GR740 microprocessor [2][3] has been manufactured with flight model availability expected for mid-2019. The GR716 microcontroller prototype silicon is scheduled for tape-out in the second quarter of 2018.

This paper contains a section for each of the components, where an overview is given of the implemented architecture, intended use, and current test and qualification status. The architectures of both the GR718B SpaceWire router and the GR740 microprocessor are already available together with extensive documentation [1][2] and this paper puts emphasis on the GR716 microcontroller, which is the most recent development of the three.

II. GR718B SPACEWIRE ROUTER

A. Component Overview

The GR718B router implements a SpaceWire routing switch as defined in the ECSS-E-ST-50-12C SpaceWire links, nodes, routers and networks standard, supporting all mandatory and optional features. Among the main features supported by the GR718B are: group adaptive routing, packet distribution, system time-distribution, distributed interrupts, port timers to recover from deadlock situations and SpaceWire-D packet truncation-based time-slot violations.

Claudio Monteleone, Roland Weigand European Space Research and Technology Centre European Space Agency Noordwijk, The Netherlands

The GR718B has 16 external routing SpaceWire ports with on-chip LVDS and two ports with LVTTL for use with offchip LVDS transceivers. The GR718B includes a number of power saving functions, which include the possibility of disabling any unused receivers/transmitters as well as clock gating of unused SpaceWire ports.

UART and JTAG interfaces, that give access to the on-chip AMBA AHB bus, are provided for configuration and debugging. SPI and GPIO interfaces are accessible through the configuration port.

B. Radiation tolerance

The results of the radiation qualification demonstrate the GR718B's tolerance of 300krad(Si) and its immunity to single event latch-up.

C. Qualification status

The screening and lot qualification of space-level GR718B SpaceWire routers has been done following the lot validation approach defined in ESCC 9000 chart F2 and chart F4 of [4], with additional tests from MIL-STD-883 test methods [5].

The first space-level batch has been manufactured, screened and lot validation tested successfully. A description of the lot validation tests has been provided herein. No failures were reported for any of the tests performed during the lot validation.

Wafer lot acceptance test of the wafers used for the manufacturing of the first space-level GR718B batch has also been successfully completed.

D. Target Technology and Package

The GR718B SpaceWire router is available in a hermetically sealed, 256 leads, ceramic package. It has been designed for operation over the full military temperature range -55° C to $+125^{\circ}$ C.

The GR718B has been manufactured in UMC's 180nm CMOS technology using Imec's "Design Against Radiation Effects" (DARE) library [6].

III. GR740 MICROPROCESSOR

A. Component Overview

The GR740 microprocessor is a quad-core 32-bit microprocessor with an integrated 2 MiB Level-2 cache. The device provides communication interfaces such as Gigabit Ethernet, MIL-STD-1553B, CAN 2.0B, PCI 2.3 and has an integrated SpaceWire router with eight external SpaceWire links.

GR740 prototype devices are currently available and a design revision, to be used for future prototypes and flight models, has been manufactured.



Fig. 1. GR740 block diagram

B. SpaceWire Router

The G740 microprocessor contains the same SpaceWire router IP core as the one used for the previously described GR718B SpaceWire router, however the implemented configuration of the router differs between the two devices. The initial specification for the GR740 was to include four separate SpaceWire codecs, including Remote Memory Access Protocol (RMAP) targets, connected to the on-chip bus. Each codec would be implemented with redundant ports providing a total of eight external SpaceWire links of which four could be concurrently utilized. With the introduction of the SpaceWire router IP, the GR740 design was changed to instead implement a router with eight external SpaceWire links. The router also provides four internal ports, referred to as AMBA ports, connected to the on-chip bus. These four AMBA ports each provide both a hardware RMAP target and a interface on the on-chip bus that is compatible with the standalone SpaceWire codecs that were initially specified to be used in the GR740. Inclusion of the router with the AMBA ports in the design allows the initial use case with four separate codecs and also enables other use cases in systems that allow the router to physically located within the microprocessor.

Production and bench tests of the SpaceWire interfaces of the GR740 successfully utilized the SpaceWire links with link rates up to and over 300 Mbit/s in the prototype devices. The flight silicon tests have not yet been performed. Static timing analysis shows that all design parameters are better or equal compared to the prototype silicon.

C. SpaceWire Debug Link

GR740 provides a dedicated debug subsystem, shown in the top right part of Fig. 1. . The debug subsystem provides

access to a processor debug support unit that allows to monitor and control execution of the four LEON4FT processor cores. The debug subsystem also provides nonintrusive access to performance counters and trace buffers present in the design.

An external debug host connects to the debug subsystem through Ethernet, JTAG or SpaceWire. The SpaceWire link is provided by a codec that is separate from the SpaceWire router implementation. In order to minimize the number of wires that need to be exposed in an integrated system it is also possible to connect the debug SpaceWire link through one of the router ports. The connection needs to be performed externally to the GR740 device and would sacrifice one of the GR740 router's external links to connect the debug subsystem to the SpaceWire network.

D. SpaceWire Time Distribution Protocol

The GR740 includes a SpaceWire Time Distribution Protocol (TDP) controller. This controller provides basic time keeping functions such as Elapsed Time counter according to the CCSDS Unsegmented Code specification. It provides support for setting and sampling the Elapsed Time counter. It also includes a frequency synthesizer with which a binary frequency is generated to drive the Elapsed Time counter. This interface also implements the SpaceWire - Time Distribution Protocol. The protocol provides capability to transfer time values and synchronize them between onboard users of SpaceWire network. The time values are transferred as CCSDS Time Codes and synchronization is performed through SpaceWire Time-Codes. The core also provides datation services.

E. Design Updates in Flight Silicon

The design revision to be used for the GR740 flight models includes a number of updates. The updates relevant in a SpaceWire context are extensions to the SpaceWire TDP controller and its integration in the design. The new features are:

- Added support for 64 Distributed interrupts
- Increment time using External enable signal
- Load Time using External enable signal
- Logic to detect Missing SpaceWire Time-code
- Latency corrected status can be write cleared
- Added additional option to synchronize time, using only SpaceWire time-codes it is possible to synchronize time.

F. Qualification status

A design revision of the GR740 to be used for flight model screening and qualification has been manufactured as of March 2018. The qualification effort is scheduled to start in the third quarter of 2018. Flight model availability is expected in 2019.

G. Target Technology and Package

The technology used is ST Microelectronics 65 nm bulk CMOS, using the C65SPACE platform. The device is available in hermetically sealed ceramic LGA625 package, also available as a CGA625.

IV. GR716 MICROCONTROLLER

A. Component Overview

The GR716 is a 32-bit microcontroller with large number of on-chip functions such as UART, SPI, ADC, DAC, I^2C , CAN and MIL-STD-1553B. The device has one SpaceWire interface that supports a 100 Mbit/s link rate.

B. Background of GR716 Development

Software based data acquisition, data processing and simple control applications are widely used in spacecraft subsystems. They allow implementation of software based control architectures that provide a higher flexibility and autonomous capabilities versus hardware implementations. For this type of applications, where limited processor performance is required, general purpose microprocessors are usually considered incompatible due to high power consumption, high pin count packages, need of external memories and missing peripherals. Low-end microcontrollers are considered more attractive in many applications such as:

- Propulsion system control
- Sensor bus control
- Robotics applications control
- Simple motor control
- Power control
- Particle detector instrumentation
- Radiation environment monitoring
- Thermal control
- Antenna pointing control
- AOCS/GNC (Gyro, IMU, MTM)
- RTU control
- Simple instrument control
- Wireless networking

In these kind of applications the microcontroller device should have a relatively low price, a low power consumption, a limited number of pins and must integrate small amount of RAM and most of the I/O peripherals for control and data acquisition (serial I/Fs, GPIO's, PWM, ADC etc.). The requirements for memory and program length are usually minimal, with no or very simple operating system and low software complexity.

C. Microcontroller Applications

Spacecraft subsystem control and monitoring of parameters such as power supply voltages, currents, pressures and temperatures are ideal applications for the LEON3FT microcontroller. Bridges between different communication standards or interface of equipment towards a higher-level controller or the central On Board Computer (OBC) are also ideal applications for the LEON3FT microcontroller.

The microcontroller can perform advanced data handling to offload any higher level controller or the central On Board Computer (OBC). By hiding the data handling details the transmitting data volume can be reduced and simplified functionalities and timing requirements are requested to the higher level controller. The LEON3FT microcontroller integrates several on-chip data bus standards, such as SpaceWire, CAN, MIL-STD-1553, I²C, SPI, UART and can easily provide data packetization for serial communication using standard protocols. The microcontroller can also efficiently replace FPGAs in accomplishing the above functionalities. Generally the FPGA implementation is faster but much more complexity and flexibility can be captured in the software of a microcontroller even with limited processing capability. The correct use of FPGAs in space applications can be complex to achieve and also cost, package size and availability of integrated analog functions can favor the use of a microcontroller with respect to FPGA.

Below are listed a number of possible microcontroller use cases and specific applications.

- Nanosatellite controller
- Instrument Control Unit
- Remote Terminal control
- Mass Memory control
- Propulsion Unit control
- Electric Motor Control

D. Microcontroller Architecture

Fig. 2. shows an overview of the mixed architecture of the GR716. The mixed architecture integrates many system functions for easy system integration e.g. power on reset and clock generation.



Fig. 2. Architecture overview

The digital system is shown in Fig. 3. and the system consists of three AMBA AHB buses, one main system bus, one debug bus and one bus for DMA traffic.

The main bus includes the LEON3FT processor connected to a shared on-chip RAM and ROM. The main bus also connects all other peripheral cores in the design as well as the external memory controllers. Several peripherals are connected through two AMBA AHB/APB bridges where the bridges are integrated with the design's DMA controller.

The debug AMBA AHB bus connects a serial UART debug communications link to the debug support unit and also to the rest of the system through an AMBA AHB bridge.

The third bus, a dedicated 32-bit Debug bus, connects a debug support unit (DSU), AHB trace buffers and several debug communication links. The Debug bus allows for non-intrusive debugging through the DSU and direct access to the complete system, as the Debug bus is not placed behind an AHB bridge with access restriction functionality.



Fig. 3. Digital architecture overview

The list below summarizes the specification for the complete system:

- System Architecture
 - Fault-tolerant SPARC V8 processor with 32 register windows, 192KiB EDAC protected tightly coupled memory and reduced instruction set
 - Double precision IEEE-754 floating point unit
 - Advanced on-chip debug support unit with trace buffers and statistic unit for software profiling
 - Memory protection units with 8 zones and individual access control
 - Single cycle instructions execution and data fetch from tightly coupled memory
 - Deterministic instruction execution and interrupt latency
 - Fast context switching (PWRPSR, AWP, Register partitioning, irq mapping, SVT, MVT)
 - Atomic operations support
 - Memories
 - 192KiB EDAC protected tightly coupled memory with single cycle access from processor and ATOMIC bit operations.
 - Embedded ROM with bootloader for initializing and remote access
 - Dedicated SPI Memory interface with boot ROM capability
 - I²C memory interface with boot ROM capability
 - 8-bit SRAM/ROM (FTMCTRL) with support up to 16 MB ROM and 256 MB SRAM
 - Support for package option with embedded SRAM/PROM.
 - Scrubber with programmable scrub rate for all embedded memories and external PROM/ SRAM and SPI memories
- System
 - On-chip voltage regulators for single supply support. Capability to sense core voltage for trimming of the

embedded voltage regulator for low power applications

- Power-on-reset, Brownout detection and Dual Watch Dog for safe operation. External reset signal generation for reseting companion chips
- Crystal oscillator support
- PLL for System and SpaceWire clock generation
- Low power mode and individual clock gating of functions and peripherals
- Temperature and core voltage sensor
- External precision voltage reference for precision measurement
- Four programmable DMA controllers with up to 16 individual channel
- Timer units with seven 32-bit timers including watchdog
- Atomic access support for all APB registers (AND, OR, XOR, Set&Clear).
- Peripheral access control
- Embedded trace and statistics unit for profiling of the system
- Peripherals
- SpaceWire with support for RMAP and Time Distribution Protocol
- Redundant MIL-STD-1553B BRM (BC/RT/BM) interface
- Multiple CAN 2.0B bus controllers
- Six UART ports, with 16-byte FIFO
- Two SPI master/slave serial ports
- SPI4SPACE. Hardware support for SPI protocol 0,1 and 2 in HW for SPI for SPI4SPACE
- Two I²C master/slave serial port
- PacketWire interface
- PWM with up-to 16 channels. PWM clock support up to 200 MHz
- Up to 64 General input and outputs (GPIO) with external interrupt capability, pulse generation and sampling
- Four single ended Digital to Analog Converters (DAC), 12-bit at 3MS/s
- Four differential or eight single ended Analog to Digital Converters (ADC) 11-bit at 200KS/s with programmable pre-amplifier and support for oversampling. Dual sample and hold circuit integrated for simultaneously sampling
- External ADC and DAC support up to 16-bit at 1MS/s I/O
- Configurable I/O selection matrix with support for mixed signals, internal pull-up/pulldown resistors
- LVDS transceivers for SpaceWire or SPI4SPACE
- Dedicated SPI boot ROM support for configuration
 Supply
- Single 3.3V±0.3V supply or separate Core Voltage 1.8V±0.18V, I/O voltage 3.3V±0.3V

E. Processor Performance and Determinism

In order to improve determinism, the LEON3FT microcontroller contains a local instruction and data static RAM with fixed response times. All EDAC units in the system have the same latency and behavior in the corrected as in the uncorrected case. This also applies to the CPU, so dynamic SEU handling schemes such as the LEON3FT pipeline restart on error options is not used.

Local instruction RAM tightly coupled to the LEON3FT CPU will be the main memory to execute the software. Due to its direct connection to the CPU, the execution of the software will be deterministic. For applications where full cycle-level determinism is not needed, it will also be possible to execute software from an external SRAM.

The local instruction memory will be implemented using dual-port RAM. The memory's second port will be connected to the main system AHB. This will allow modifying of the local instruction RAM without the intervention of the CPU. The contents of this memory will be protected against SEU errors with EDAC and scrubbing.

If the DMA peripherals and the processor are connected to a shared single-port memory, or to a memory via the same bus, and try to simultaneously access the shared resource then the DMA activity will have an effect on the execution time. On the other hand DMA activity will have no impact on SW execution time by using a dual-port on-chip data RAM and a separate bus for the DMA peripherals. This means that there is a separate access path for the CPU core to local instruction and data RAMs that is unaffected by concurrent DMA activity.

For applications demanding determinism on nested interrupts, a special interrupt-handling scheme will be implemented in software where nested interrupts are allowed to occupy one additional register window. The number of levels of nested interrupts that can be handled without additional timing penalty depends on the complexity of the software implementation.

In the architecture, deterministic interrupt latency is achieved by:

- Running software (including interrupt handlers) from local RAM and accessing any data needed during the interrupt handling through port separate from AMBA ports.
- Adapting the register window usage (using a flat model) structure to avoid unexpected window over/underflow traps. This is done in the compiler and application code, and most OS code does not need modification.
- The alternate window pointer (AWP) feature from the SPARC V8E extension to allow window over/underflow handlers to run with traps enabled.
- Register file partitioning to allow partitioning of the register file (the windows) to different "contexts". Contexts can for example be threads to speed up context switching and/or interrupt contexts to dedicate windows to ISRs.

F. Utilizing the Flat Register Model

Very low interrupt response time can be achieved by utilising LEON3FT microcontrollers large number of register windows. Low interrupt response time can be achieved by using a flat, or single-register, register window model. The flat register window model eliminates window underflow/overflow-trap and the number of cycles executed with trap disables.

A real-time application compiled with the single register window model using GCC does not issue any SAVE or RESTORE instructions so it executes inside a single window. When a trap is taken (interrupt trap), the current window is decremented with one.

Each unique trap handler is to execute each level of the interrupt nest hierarchy in its own window. Since the flat ABI preserves local and input registers as needed, there is no need for the interrupt trap handler to store these. The output registers become the next interrupts input registers and are thus "protected" by the ABI. Only the global registers have to be stored/restored when jumping between interrupt nest levels. There is room to temporarily store the global registers in the local registers of the trap window.

Examples provided with software environment version 2.0.2 demonstrates that the total latency from interrupt assert to interrupt service routine (ISR) can be as low as less than 28 clock cycles.

Benefits from using flat register model:

- Interrupt acknowledgement to ISR is 22 cycles (constant)
- Interrupt exit is 15 cycles
- No registers stored to memory in interrupt trap
- No registers loaded from memory in interrupt trap (except to get ISR handler)
- Supports interrupt nesting
- No application specific considerations

Limitations of flat register model:

- If SVT is used, then 11 cycles have to be added for the ack to ISR cycles
- Requires one register window per nested interrupt request level

G. Programmable DMA Controller

Cobham Gaisler has developed a DMA controller able to preform concurrent programmable sequences of data transfers between any on-chip peripherals in the AMBA address space. The DMA controller is able to transfer data both between peripherals, between peripherals and memory and between memory areas. If the accessed memory is internal or external does not matter, as long as the memory is mapped into AMBA address space reachable from the AHB bus where the core is mapped.

The DMA controller has been specifically designed to offload the CPU and provide DMA capabilities to peripherals that do not have an internal DMA engine. The CPU is offloaded by the fact that the peripheral event is directly routed to the DMA controller. By routing events directly to the DMA controller or even directly between peripherals, these interrupts are in effect offloaded from the CPU. These reduce also the number of concurrent interrupts the CPU must handle and that may erode the system determinism.

H. SpaceWire implemetation

SpaceWire has a key role in the foreseen GR716 applications due to the ease of using SpaceWire RMAP for uploading software to the microcontroller, reducing the need for external memory components. By putting the GR716 in remote boot mode it is possible to connect to the device using SpaceWire RMAP, upload software and then start processor execution.

The GR716 implements the GRSPW2 SpaceWire controller from Cobham Gaisler. Through this, the same interface available in previous generation microprocessors such as the LEON3FT dual-core GR712RC and the LEON3FT UT700 device is available in the GR716 component. The SpaceWire codec has been implemented with redundant port functionality. This provides two ports, of which one can be used at a time, one with on-chip LVDS and one provided with signals at LVTTL level.

I. Pin-Multiplexing

Because of the small package and high number of interfaces, the functionality of the pins must be configurable and the pins must be shared between several peripherals. The number of configurable user pins has been chosen to be 64.

J. Clocking, Reset and Maximum Operating Frequency

The maximum operating frequency for the AMBA system is 50 MHz. The device can have separate clock signal inputs for system, SpaceWire and MIL-STD-1553B interfaces. The clocks signals can also be derived from single source via clock multipliers and dividers inside the device.

In order to avoid problems with reset sequencing, the device has one single reset input that is sequenced internally to provide reset signals to the different clock domains within the device.

K. Support for PROM-Less Applications

The device provides an easy access for systems that want to avoid having a boot-PROM connected to the device and prefer to upload software remotely.

As previously described, the device can be remotely configured via SpaceWire. The same functionality is also available via the SPI, UART and I^2C interfaces.

L. Target Technology and Package

The technology used is UMC 180 nm, using the DARE library[6] from IMEC, and the package is a 132 pin CQFP.

V. SOFTWARE SUPPORT

The components described in this paper are all supported by operating systems, simulators and toolchains provided by Cobham Gaisler.

VI. CONCLUSION

The GR718B SpaceWire router, GR740 microprocessor and GR716 microcontroller are standard products from Cobham Gaisler at different stages in their product life cycle. All of the components implement SpaceWire support as a key feature where the router functionality enables SpaceWire networks to which the microcontroller and microprocessor devices can connect.

The GR716 is a 32-bit microcontroller with one SpaceWire interface that supports a 100 Mbit/s link rate. The GR740 features a SpaceWire router with eight external links. Both devices have a compatible on-chip bus interface to the SpaceWire codecs where traffic can be handled either by hardware RMAP targets or through software running on the device controlling the on-chip interfaces using memory-mapped IO and in-memory descriptor structures.

SpaceWire has a key role both the microprocessor and microcontroller case both as general-purpose communication links and due to the ease of using SpaceWire RMAP for uploading software, reducing the need for external memory components.

ACKNOWLEDGMENT

The GR740, GR716 and GR718B developments have been developed in activities initiated or supported by the European Space Agency.

REFERENCES

- [1] Cobham Gasisler, "GR718B Data Sheet and User's Manual", http://www.gaisler.com/GR718B [online]
- [2] Cobham Gasisler, "GR740 Data Sheet and User's Manual", http://www.gaisler.com/GR740 [online]
- [3] European Space Agency, "GR740: The ESA Next Generation Microprocessor (NGMP)", http://microelectronics.esa.int/gr740/index.html [online]

http://incroelectromes.esa.int/gr/40/index.ntin [onnie]

- [4] ESCC Generic Specification No. 9000. Integrated circuits: Monolithic and multichip microcircuits, wire-bonded, hermetically sealed and flip-chip monolithic microcircuits, solder ball bonded, hermetically and non-hermetically sealed. February 2016. Available on-line: https://escies.org
- [5] Department of Defense Standards, MIL-STD-883 Revision K. Test method standard microcircuits. April 2016.
- [6] S. Redant, R. Marec, L. Baguena, E. Liegeon, J. Soucarre, B. Van Thielen, G. Beeckman, P. Ribeiro, A. Fernandez-Leon and B. Glass. Radiation test results on first silicon in the Design Against Radiation Effects (DARE) library, IEEE Transactions on Nuclear Science, VOL. 52, NO. 5, October 2005.

Components 2 (Long)

SpaceFibre Interface and Routing Switch IP Cores

SpaceFibre, Long Paper

Albert Ferrer Florit, Alberto Gonzalez Villafranca STAR-Barcelona S.L. Sant Cugat, Spain albert.ferrer@star-dundee.com

Abstract—SpaceFibre is a technology specifically designed for use onboard spacecraft that provides point to point and networked interconnections at Gigabit rates with Quality of Service (QoS) and Fault Detection, Isolation and Recovery (FDIR). SpaceFibre is backwards compatible with SpaceWire, allowing existing SpaceWire equipment to be incorporated into a SpaceFibre network without modifications at packet level.

In this work we present the family of SpaceFibre IP Cores developed by STAR-Dundee. It is composed of three different IPs: the Single-Lane Interface, the Multi-Lane Interface and the Routing Switch. The IP Cores are fully compliant with the SpaceFibre standard and have been carefully implemented to optimise their performance and minimise their footprint on radiation-tolerant FPGAs (e.g. RTAX, RTG4, BRAVE or Virtex-5QV) and ASICs. They have also been validated on commercial FPGAs (e.g. Igloo2, Spartan, Virtex, Kintex, etc.).

The Single-Lane Interface IP offers in a compact design (~3% of the RTG4/Virtex-5QV) the maximum possible line rates provided by embedded or external transceivers (i.e. 3.125 Gbps in RTG4, 4.25 Gbps in Virtex-5QV and 2.5 Gbps in RTAX using the TLK2711-SP transceiver). The Multi-Lane Interface IP allows much higher data rates and adds all the advantages of combining multiple lanes without multiplying the resources required (e.g. ~5-6% for 3 lanes in RTG4/Virtex-5QV). The SpaceFibre Routing Switch IP Core is a scalable, fully configurable non-blocking router, allowing to select the number of virtual channels and ports. This routing switch implements path and logical addressing, group adaptive routing, virtual networks, time distribution and message broadcast. A router of 4 ports each with 4 virtual channels typically requires less than 20% of an RTG4, including the SpFi interfaces.

The IP Cores presented in this article provide the building blocks for creating the next generation of onboard networks with in-built QoS and FDIR mechanisms, and are currently being implemented in several missions and products all over the world. We analyse the performance and capabilities of the different IP Cores, and discuss the resources required depending on several parameters such as the number lanes, ports, virtual channels and virtual networks.

Index Terms—SpaceFibre, Single-Lane, Multi-Lane, Routing Switch, IP Core, FPGA, Radiation Tolerant, RTG4, Virtex-5QV, Networking, Spacecraft Electronics Steve Parkes, Chris McClements STAR-Dundee Ltd. Dundee, UK

I. INTRODUCTION

SpaceFibre (SpFi) [1-2] is a new technology for use onboard spacecraft that provides point-to-point and networked interconnections at Gigabit rates with QoS. SpFi interoperates seamlessly with a SpaceWire (SpW) [3-4] network over virtual channels (VCs), as it uses the same packet definition. It provides broadcast capabilities and is able to operate over a copper or fibre optic communication medium. SpFi will be released as an ECSS standard later this year.

New generation payloads, such as SAR and multi-spectral imaging instruments, require the use of multiple parallel highspeed links to fulfil the increasing bandwidth requirements. To accommodate these needs, SpFi supports multi-lane operation, thus allowing data to be sent over several individual physical lanes to enhance throughput and robustness. The multi-lane operation allows much higher data rates through lane aggregation, supporting any number of lanes (up to 16) and unidirectional operation. This effectively multiplies the throughput of the interface by combining several lanes into a single link. Furthermore, when a lane fails the multi-lane mechanism supports hot redundancy and graceful degradation by automatically spreading traffic over the remaining working lanes. Thus multi-lane provides all the advantages of combining multiple lanes without multiplying the resources required.

The SpFi Network layer is responsible for transferring packets over a SpFi link or network. The information to be sent is packaged in the same format as SpW: <Destination Address> <Cargo> <End of Packet Marker>. It uses the same routing concepts as SpW including both path and logical addressing. The SpFi Network layer defines the concept of Virtual Network (VN). VNs are built from the interconnection between VCs of different ports. These VNs enable the creation of highly flexible SpFi routing switches comprising a number of SpFi interfaces and a fully configurable, non-blocking, high performance, routing switch. This routing switch can theoretically support an arbitrary number of VNs, each effectively behaving like independent SpW networks capable of working at multi-Gbps rates.

This paper describes the SpFi Single-Lane IP Core features and performance in section II. Section III analyses the SpFi Multi-Lane IP. Section IV describes the characteristics of the SpFi Routing Switch IP. Finally, conclusions are presented in section V.

II. SPACEFIBRE SINGLE-LANE INTERFACE IP CORE

The STAR-Dundee SpaceFibre Single-Lane Interface IP Core is compliant with the SpFi standard with the exception of the Multi-Lane layer, which has been added in a separate IP (presented in Section III).

A. Characteristics

This IP has been designed to minimise as much as possible the designer effort of adding the SpFi interface into a design. It is provided with a protocol agnostic data interface, so that no prior knowledge of the SpFi standard is required. Simple data interfaces based on standard 32-bit input and output FIFO interfaces are used. Specifically, they follow the AXI4-Stream protocol [5], which is an industry standard commonly used. This AXI4-Stream interface allows using independent userdefined read and write clocks. Inside them the clock synchronisation between the user local clock domains and the SpFi IP clock domain is done.

The IP can be configured using generics. Different properties can be configured:

- The type of transceiver interface
- The number of VCs
- The size of the VC buffers
- The type of Management interface

There are different high-speed transceiver interface options that provide simple set of signals to be directly connected to the selected transceiver. However, for user convenience different wrappers are supplied for the different transceiver interfaces. For example, old FPGA technologies require external transceivers. In the IP there is a dedicated TLK2711-SP (Wizardlink) [6] interface with all the required signals readily available at the top level. Regarding newer FPGA technologies, there is a specific interface for the RTG4 SerDes, with the 8B/10B encoding (20-bit interface), clock correction and symbol alignment done inside the IP. In this case, three clocks are required by the IP. If we define the lane speed as Ls then two clocks with frequencies of $L_S/20$ and $L_S/40$ are needed for transmission and general operation (e.g. 156.25 & 78.125 MHz @ 3.125 Gbps). The slowest clock is the main clock of the IP and is used by most of the logic. In addition, a recovered clock working at $L_s/20$ is required in reception.

In the case of Xilinx devices (e.g. Virtex-5QV), a simpler interface (32+4 bits) is available that benefits from the additional capabilities offered by their in-built transceivers. The 8B/10B encoding, clock correction and symbol alignment are done inside the Xilinx transceiver. Only a single clock is then required by the IP, with the exception of the AXI clocks of the VC interfaces. This clock operates at a frequency of $L_s/40$ (e.g. 78.125 MHz @ 3.125 Gbps).

The QoS is independently and dynamically configurable for each VC. Three different mechanisms can be configured and combined: scheduling, priority and bandwidth reservation. The FDIR mechanisms automatically recover from transient and persistent errors on the SpFi link. A transient error takes less than 2 µsec to recover and does not affect the user data rate thanks to the embedded buffering inside the IP. Other protections against errors include data and broadcast babbling idiot protection. Tests have been done to validate that data integrity is maintained for Bit Error Rates (BER) better than 10^{-5} . The lane is automatically disconnected when the BER is worse than 10^{-5} to prevent a potential protocol breakdown.

The IP has been optimised for low latency operation. Moreover, it offers an 80-bit interface that makes use of the broadcast message mechanism in SpFi to deliver ultra-low latency short messages. These messages present less than 400 ns of guaranteed latency for a few meters of cable.

A simple management interface allows real time configuration of the control parameters and access to the IP status. Furthermore, this interface also includes optional statistics and debug signals. Two different types of management interfaces can be selected: APB bus or signal bus. Having independent signals for each status and configuration field is useful when implementing the IP in an FPGA design that needs direct access to the IP. The alternative is to access to all these fields over an APB bus. This greatly simplifies the interface for designs that use a CPU or want a centralised access point to several interfaces, for example. The APB bus also comes with an independent clock for convenience, and manages internally the clock synchronisation with the clocks of the SpFi IP.

Finally, power management options have been considered. For example, there is the possibility of start one end of the link in a low-power mode waiting for the other end to become active. A block diagram of the interconnection of the IP Core is shown in Fig. 1.



Fig. 1. SpaceFibre Single-Lane interconnection block diagram

B. Performance

The SpFi Single-Lane IP design has been validated in the main FPGA families relevant to space, such as Microsemi RTG4 [7], and Xilinx Virtex-5QV [8] and RTAX. It offers in a compact design the maximum possible line rates in radiation-tolerant FPGAs provided by embedded or external transceivers (i.e. 3.125 Gbps in RTG4, 4.25 Gbps in Virtex-5QV and 2.5

Gbps in RTAX using the TLK2711-SP device). The IP will also be compatible with the NanoXplore BRAVE Large FPGA [9], and can be implemented in commercial FPGA families such as SmartFusion, Spartan, Kintex, Zynq, etc.

The IP has been extensively tested in the RTG4 and Virtex-5QV to guarantee timing closure for the whole temperature and voltage range (i.e. fast & slow corners) required by the space devices, including the EDAC in the memories and the Single Event Transient (SET) filtering enabled. These radiation mitigation techniques reduce the maximum speed of the designs and can sometimes give problems to meet the targeted speeds. Additionally, the IP has been tested under radiation in an RTG4. The information gathered during the test campaign has allowed for further refining the operation under radiation of the IP and the associated reference design in the RTG4. This IP Core is also ready for ASIC implementation. It has been used in the new RC64 many-core DSP ASIC developed by Ramon Chips. This processor features 12 SpFi ports with line rates of up to 6.25 Gbps [10].

The RTG4/Virtex-5QV resources required by the SpFi interface including transmit and receive FIFOs are detailed in Table I for different numbers of VCs. Note than adding a VC to the design has a small impact in the overall resource usage.

TABLE I. SPFI SINGLE-LANE RESOURCE USAGE

	RTG4			Virtex-5QV		
	LUT	DFF	LSRAM Block	LUT	DFF	RAM Block
1 VC	3944	2818	4	2750	2365	4
IVC	2.6%	1.9%	1.9%	3.4%	2.9%	1.3%
2 VCs	4454	3197	6	3138	2596	6
2 105	2.9%	2.1%	2.9%	3.8%	3.2%	2.0%

III. SPACEFIBRE MULTI-LANE INTERFACE IP CORE

Multi-lane is an optional capability of a SpFi link defined in the Multi-Lane layer of the SpFi protocol stack. As shown in Fig. 2, the Multi-Lane layer is defined between the Data Link layer and the Lane layer implemented for each available lane.

The Data Link layer provides QoS and flow control for a SpFi link. It frames the information to be sent over the link and also provides error recovery capabilities, detecting any frames or control words that go missing or arrive containing errors and resending them. On the other hand, the Lane Layer establishes a connection across a SpFi lane and ensures that bit, symbol and word synchronizations are achieved and that the two ends of the lane are both ready to send and receive data. The Lane Layer also encodes data and control words into 8B/10B symbols, sends and receives symbols over the lane and decodes the received symbols into data and control words. The Multilane layer coordinates the operation of multiple lanes as a single SpFi link, providing higher data throughput and redundancy. Because the logic that initializes a lane and monitors its status is located below the multilane layer, each lane can be initialized and operated independently of each other.



Fig. 2. Multi-Lane layer in the SpaceFibre layer stack

This architecture also supports the implementation of graceful degradation, which means that in the event of one or more lanes failing, traffic is spread over the remaining working lanes automatically. When combined with the Data Link layer QoS, the bandwidth allocated to lower priority VCs is reduced when required to ensure that most important information gets through and deterministic traffic is maintained.

A. Characteristics

The SpFi Multi-Lane IP core has been designed to be easy to use, with minimum configuration signals. On top of the features of the Single-Lane IP Core already mentioned in Section II, the Multi-Lane IP also features the additional capabilities specifically related to the Multi-Lane layer.

The number of lanes is fully configurable, with any number of lanes supported (up to 16). Each lane can independently be selected as uni/bidirectional and hot/cold redundant. Single-Lane SpFi implementations must be bidirectional even if the end-user data flow is unidirectional, because of the feedback required by the protocol. However, in a Multi-Lane implementation only one bidirectional lane is enough for protocol related information. Therefore, other lanes can be unidirectional to save power and mass in asymmetric data flows.

In the 4-lane configuration example of Fig. 3, bidirectional lane 2 can be set to a unidirectional lane for power saving reasons. Unidirectional Lane 4 can be enabled when one lane fails or a higher data rate is required. It is also possible to send data using all four lanes and add an additional one configured as a hot redundant lane, which only sends data when another lane fails.



Fig. 3. Unidirectional and bidirectional lanes interconnection example

The IP Core has been designed to fully support the redundancy capabilities offered by the Multi-Lane. Hot redundant lanes recover not only from transient errors like in the Single-Lane case, but also from lane failures in less than $2 \ \mu s$ without user intervention and without any data loss. This time is close to the round trip delay of the lane, so short than the data flow of the user is not affected when a lane fails, as the data is internally buffered during the time it takes to resume sending data. In case of lane failure when not using a hot redundant lane, there is an automatic graceful degradation of the link bandwidth, with higher priority VCs being less affected. The QoS mechanism ensures that the most important data is sent first. If a cold redundant lane is available it will be activated in less than $20 \ \mu s$, providing warm redundancy.

Bandwidth overprovision and dynamic power management are also possible. These capabilities are very useful for space applications where strict power constrains and a high level of reliability is required on the harsh space environment.

The width of the AXI4-Stream interface for the VCs is configurable in multiples of the SpFi word size (32-bits). This allows supporting slower user clocks and still being able to send or receive data at the maximum speed over a single VC. For example, a 64-bit width can be used to operate a link with 2 lanes at the same clock speed of a single-lane link.



Fig. 4. SpFi Multi-Lane IP interconnection test between RTG4 and Spartan FPGAs

B. Performance

Table II provides the resource usage for two radiation hardened FPGAs, Microsemi RTG4 and Xilinx Virtex-5QV for different number of lanes and VCs. Individual lanes can operate up to

3.125 Gbps, with aggregate rates of up to 6.25 Gbit/s using 2 lanes or up to 12.5 Gbit/s using 4 lanes, for example. This Multi-Lane IP has been tested in the RTG4 and Virtex-5QV to guarantee timing closure with memories using EDAC and the Single Event Transient (SET) filtering enabled in worst case (slowest) scenario. It has also been tested under radiation running in an RTG4, just like has been done with the Single-Lane. Fig. 4 shows the Multi-Lane IP integrated in an RTG4 transferring data to a PXI board with a Xilinx commercial device.

TABLE II. SPFI MULTI-LANE RESOURCE USAGE

	RTG4				Virtex-5QV		
	LUT	DFF	LSRAM Block	µSRAM Block	LUT	DFF	RAM Block
2 Lanes	6794	4534	8	3	3858	3938	8
1 VC	4.5%	3.0%	3.8%	1.4%	4.7%	4.8%	2.7%
2 Lanes	7736	5285	12	3	4503	4382	12
2 VCs	5.1%	3.5%	5.7%	1.4%	5.5%	5.3%	4.0%
3 Lanes	8771	5593	8	4	4770	4849	8
1 VC	5.8%	3.7%	3.8%	1.9%	5.8%	5.9%	2.7%
3 Lanes	9691	6346	12	4	5416	5226	12
2 VCs	6.4%	4.2%	5.7%	1.9%	6.6%	6.4%	4.0%

IV. SPACEFIBRE ROUTING SWITCH IP CORE

The router architecture is built around a non-blocking routing switch matrix with a configurable number of ports. Each port implements a number of VCs. Each VC has an associated VN number. The switch matrix interconnects one or more VCs with the same VN number, but each of these VCs must be located in a different port.

Packets belonging to different VNs never interfere with one another and do not impact the throughput and latency within the routing switch matrix. On the other hand, when multiple packets in the same VN need to be transferred to the same output port, packet-by-packet, round-robin arbitration is performed, similarly to a SpW router.

A. Characteristics

The STAR-Dundee SpaceFibre Routing Switch Core (SpFi Router for short) IP is a scalable, fully configurable nonblocking router. The IP is very flexible, allowing to select the number of VCs and ports, among other options, using generics. This router implements path and logical addressing, group adaptive routing, VNs, time distribution and message broadcast. In addition, it also fully supports the QoS and FDIR capabilities native to SpFi.

The IP Core has been optimised to work with the existing high-performance radiation-tolerant FPGAs using an arbitrary number of SpFi and internal ports. Each port supports up to eight VCs, each with its own QoS parameters. Furthermore, the Router includes an optional configuration port with RMAP protocol. The maximum number of VNs currently supported is 64, but each of these VNs is completely flexible: any VC of any port can be configured to any VN. The IP Core can also be implemented in ASIC technologies. The high flexibility of the SpFi Router IP Core ensures different user needs can be accommodated with ease.

The Router IP presents a deterministic low latency switching. Round-robin packet arbitration can only occur within each VN. When arbitration is required, it only takes place when two or more VCs request to access to the same output port within the same VN. The Router offers two configurable time-outs. These two timers are required to prevent the blockage of a VN in cases when the source or the sink stall in the middle of a packet, or when there is a babbling node. When timing out the router performs automatic packet spilling of the packet causing the blockage.

The VNs can be configured statically during FPGA programming using VHDL constants, or they can be dynamically modified by the user using logic connected to the configuration port or the RMAP protocol, which can be accessed over one of the ports of the Router.

B. Performance

The Router IP has been validated in commercial FPGAs, e.g. Xilinx Spartan (Fig. 4) or Kintex families, and in spacegrade FPGAs such as the RTG4 (Fig. 5). It supports lane rates up to 3.125 Gbit/s in RTG4 or Virtex-5QV, with guaranteed timing closure in RTG4 or Virtex-5QV with EDAC and SET filter enabled and worst-case conditions.



Fig. 4. 9-port SpFi Routers mounted in a cPCI rack forming a SpFi network



Fig. 5. RTG4 PXIe board used to validate the SpFi Router

The actual resources used by the complete SpFi Router design depend on the FPGA and the specific user configuration. In the case of the RTG4, Table III presents the resource usage for two different Router sizes, and for two different types of VN configuration. The values reported already include the SpFi Single-Lane IP Cores used by each of the ports and an additional RMAP configuration port. The Static VNs indicated in the table consists of a version in which the VNs are pre-defined during the design phase and hardcoded into the Router design. This can be enough for certain usecases in which the VNs are not going to change. Note that the values for the Static VNs column have been obtained for the most demanding VN configuration. Therefore, the numbers can be further reduced if simpler or fewer VNs are required. On the other hand, if a fully-configurable set of VNs is required, then the values of the Dynamic VNs column applies. In this case the VNs can be dynamically modified over RMAP through the configuration port.

TABLE III. SPFI ROUTER RESOURCE USAGE IN THE RTG4 FPGA

	Static Virtual Networks			Dynamic Virtual Networks			
	LUT	DFF	LSRAM Block	LUT	DFF	RAM Block	
4 Ports	30990	17735	30	58226	34371	46	
4 VCs	20.4%	11.7%	14.4%	38.4%	22.6%	22.0%	
9 Ports	64085	35384	80	98339	57099	116	
4 VCs	42.2%	23.3%	38.3%	64.8%	37.6%	55.5%	

V. CONCLUSIONS

The STAR-Dundee SpaceFibre Single-Lane, Multi-Lane and Routing Switch IP Cores have been designed to be easy to implement in radiation-hardened FPGAs. In this article we have detailed the performance and capabilities of the different IP Cores, and discussed the resources required depending on several parameters such as the number lanes, ports, VCs and VNs.

It is possible to integrate a SpFi Single-Lane interface inside currently available radiation-hardened FPGAs by using only a few percentage (~3%) of the device resources while getting the maximum available speed out of their integrated transceivers. The multi-lane capability dramatically increases the data throughput of SpFi and the addition of multiple lanes provide hot and warm redundancy, or graceful degradation of the link bandwidth when no redundant lanes are available. The SpFi Multi-Lane IP Core provides all these new multi-lane capabilities to RTG4 and Virtex-5QV FPGAs with low resource usage and high performance. Therefore, if more bandwidth or additional robustness is required out of a SpFi link, the Multi-Lane IP is available at the cost of a small increase in the resource usage (e.g. 5-6% for three lanes). Finally, the SpFi Routing Switch can also be integrated in space-grade FPGAs with a size greatly varying depending on the number of VCs and ports, and whether Static or Dynamic VNs configuration is requested.

All the IP Cores have been verified in simulation and subsequently validated in hardware prototypes. Both commercial and the main radiation-hardened FPGAs have been used for these validation activities, ensuring full compatibility and defining an easy adoption path for this technology.

All the STAR-Dundee family of SpFi IP Cores come with a reference design for RTG4 (Libero), Virtex-5QV (ISE) and Kintex-7 (Vivado) that can directly be implemented in the FPGA to assist the end-user and allow an easy adoption. A comprehensive end-user test bench for ModelSim/Questa simulators is also provided, which can be used as a reference for test integration. Finally, a Bus Functional Model (BFM) is included to assist in system-level design validation activities.

Together, these IPs provide the building blocks for creating the next generation of onboard networks, and are currently being implemented in FPGA and ASIC designs by different missions and products all over the world.

ACKNOWLEDGMENT

The research reported in this paper has been supported in part by the following organisations and contracts:

- The European Space Agency under ESA contract numbers 4000102641 (SpaceFibre Demonstrator) and 17938/03/NL/LvH (SpaceFibre), and Airbus DS subcontract number G010003963 (NGMMA).
- The European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement numbers 263148 (SpaceWire-RT) and 284389 (VHiSSI).
- The UK Space Agency and CEOI-ST under University of Leicester contract number RP10G0348A02 (SUNRISE), RP10G0348B206 (SpaceFibre-VPX) and RP10G0348A207 (SpaceDSP).
- The Centre for Earth Observation Instrumentation under University of Leicester contract numbers RP10G0327D13 (LOCUS Elegant Breadboard).

REFERENCES

- S. Parkes, A. Ferrer Florit and A. Gonzalez Villafranca, "SpaceFibre Standard", Draft K2, University of Dundee, December 2017.
- [2] S. Parkes et al, "SpaceFibre: Multi-Gigabit/s Interconnect for Spacecraft On-board Data Handling", IEEE Aerospace Conference, Big Sky, Montana, 2015.
- [3] ECSS Standard ECSS-E-ST-50-12C, "SpaceWire, Links, Nodes, Routers and Networks", Issue 1, European Cooperation for Space Data Standardization, July 2008, available from <u>http://www.ecss.nl</u>.
- S. Parkes, "SpaceWire Users Guide", ISBN: 978-0-9573408-0-0, STAR-Dundee, 2012, available from <u>https://www.star-dundee.com/knowledge-base/spacewire-users-guide</u> (last visited 1st May 2018).
- [5] ARM, "AMBA AXI Protocol Version 2.0 Specification", ARM, Ed., 2010.
- [6] Texas Instruments Inc., "1.6-Gpbs to 2.5-Gbps Class V Transceiver," Texas Instruments Inc., 2018, available from <u>http://www.ti.com/product/tlk2711-sp</u> (last visited 1st May 2018).
- [7] Microsemi, "Product Brief RTG4 FPGAs PB0051", Microsemi, 2017, available from <u>https://www.microsemi.com/document-portal/doc_download/134430-pb0051-rtg4-fpgas-product-brief</u> (last visited 1st May 2018).
- [8] Xilinx, "Radiation-Hardened, Space-Grade Virtex-5QV Family Overview DS192". Xilinx, Oct. 11, 2018, available from <u>https://www.xilinx.com/support/documentation/data_sheets/ds1</u>
 <u>92 V5QV Device_Overview.pdf</u> (last visited 1st May 2018).
- [9] J. Le Mauff, "From eFPGA cores to RHBD SoC FPGA", SpacE FPGA Users Workshop, 4th edition, ESA/ESTEC, Noordwijk, 2018.
- [10] R. Ginosar, P. Aviely, T. Israeli and H. Meirov, "RC64: High Performance Rad-Hard Manycore", IEEE Aerospace Conference, Big Sky, Montana, 2016.

Photonic Transceivers for SpaceFibre Datalinks

Components Session, Long Paper

Ronald T. Logan Jr. Glenair Inc. Glendale, California, USA rlogan@glenair.com

Abstract — The SpaceFibre datalink standard includes fiber-optic physical layer that requires active a optoelectronic transceivers and fiber-optic cables. It also envisions the use of active optical cables (AOCs). In this paper, we first review the advantages in mass, loss and bandwidth of fiber optic links compared to coax for the SpaceFibre physical layer. We then report the results of ongoing development of commercial-off-the-shelf photonic transceiver and active-optical-cable components designed to meet the SpaceFibre physical layer requirements. Results are presented detailing the construction of AOCs, as well as radiation exposure tests (gamma, neutrons and protons) for these modules. The results show that these modules meet or exceed many of the requirements of spacecraft applications, and indeed they are being considered for or deployed on various spaceflight applications.

Index Terms — Relevant indexing terms: SpaceWire, SpaceFibre, Spacecraft Networking, Spacecraft Electronics, Spacecraft Photonics.

I. INTRODUCTION

The emerging SpaceFibre standard for spacecraft data networking defines options for the use of either high-speed serial copper coax cables or fiber optic transmission links between avionics modules and subsystems on spacecraft at serial data rates up to 10 Gbps [1]. Optical fiber is an ideal medium for high-speed signal transmission on space platforms, since optical fiber cables support data rates up to many 10s of Gbps, are much lighter and smaller than copper coax cables of equivalent bandwidth and length, and are immune to radiofrequency (RF) interference from adjacent cables, so therefore require no heavy RF shielding. Ground-loop issues are also reduced due to the natural electrical isolation provided by nonmetallic fiber-optic cable assemblies.

Fiber optic cable solutions exist, however, the availability of suitable photonic transceiver components for space applications has not been widespread. The major

manufacturers in the photonics industry are typically not able or willing to address the highly-specialized requirements, long design cycles, extreme environmental robustness, ultra-high reliability, traceability, radiation tolerance and small, inconsistent production volumes encountered with space applications. The development of suitable photonic transceiver hardware is typically beyond the engineering or budget capacity of most spacecraft programs. We believe this combination of factors has limited the adoption of photonic links on spacecraft, while multi-gigabit links have proliferated in non-space aerospace applications.

We therefore undertook development of photonic transceivers and Active Optical Cables (AOCs) designed to address the emerging aerospace data transmission requirements between avionics modules onboard spacecraft continue to increase, driven by the use of processors with high-speed serial data I/O to support the growing data requirements of advanced sensor systems and increased bandwidth of communications switches and satellite communications terminals.

In a previous paper [2], we discussed the development and test results of photonic transmitters, receivers and transceivers for aerospace applications, and highlighted the challenges with spacecraft transceiver design. In this paper we will first briefly review the benefits in terms of size, mass and bandwidth afforded by utilizing optical fiber instead of copper cabling for multi-gigabit data networks on spacecraft. We will then describe the design of rugged photonic AOC components for aerospace applications and then summarize the results of gamma, neutron and proton radiation testing performed on the transceiver components.

II. COMPARISON OF COPPER AND FIBER OPTIC LINKS FOR SPACEFIBRE PHYSICAL LAYER

In this section we illustrate some of the advantages of fibre optic cabling compared to copper-based coax cables for spacecraft interconnects by way of a practical example. We consider a digital serial datalink operating at 1 Gbps up to 10 Gbps between two modules on a spacecraft using the SpaceFibre data networking protocol standard. In this link, the serial differential digital current-mode-logic (CML) signals are to be sent in both directions between Module 1 and Module 2. As illustrated in the block diagram of Figure 1, and according to the SpaceFibre draft standard, this can be accomplished in two ways:

1. Using four copper coax cables (two coaxes for each of the 100-ohm differential signal pairs), or

2. Using two fiber optic links (one fiber for each signal path).



Figure 1. Comparison of SpaceFibre physical layer interconnect options: a. four coax cables, b. two optical fibers and two photonic transceivers.

In the case of the coax link, it is assumed that the signal level introduced to the coax corresponds to the standard for CML at 800-1200mV, with minimum input signal level of 200 mV required for the CML receiver at the output of the coax. This corresponds to a maximum permissible loss of 12 dB over the frequency band corresponding to the digital signal frequency content, specified as 0 GHz for a 10 Gbps signal. This consideration drives the choice of coax cable type to minimize mass according to the length of the link in question. For SpaceFibre cabling as specified in the draft standard (Axon AW 2.2), the mass of the four cables is 48 g/m, the diameter of each cable is 2.4mm, and the loss at 10 GHz is 3 dB/m. For low-loss space-grade cable (such as Gore Type-21), the mass is higher at 228 g/m, the diameter is 4.8 mm, but the loss is lower at 0.3 dB/m.

By comparison, the fiber optic links require two fibers per link with mass of 10 g/m, diameter of 2 mm per jacketed simplex fiber, and the loss of the link is fully compensated by the photonic transceiver electronics to generate a fullamplitude CML signal at the end of the link. It is also possible to configure both fibers into one jacket, since the diameter of the optical fiber buffer is just 250 microns. Using two separate optical fiber jackets is a "worst case" assumption, and the mass could be reduced further if both fibers were contained in a single jacket. The photonic transceivers required at each end of the link each have mass of 5 g, and circuit board footprint of approximately 2 cm x 2 cm. Figure 2 illustrates the relative size and mass of the three types of cable. Figure 3 illustrates the cable loss vs length at 10 GHz.





Figure 3. Loss in dB at 10 GHz for SpaceFibre Type-A coax, Gore Type-21 low-loss coax, and fiber optic link.

The SpaceFibre specification permits up to 2 dB loss per connector at each end of the link. Therefore, we must add 4 dB to the losses shown in Figure 3 for the SpaceFibre and Gore coax link. This means that the SpaceFibre coax link has a maximum length of approximately 3.5 meters before it reaches the maximum permissible loss of 12 dB. The Gore Type-21 cable would support much longer links in excess of 20 meters.

The fiber optic link is essentially "lossless", due to the regeneration of the signal in the receiver circuit to a full CML output level of 800 mV p-p. This holds true provided the optical power incident on the photodiode is above the

minimum sensitivity threshold for error-free transmission. The SpaceFibre draft standard specifies this minimum optical level to be -16 dBm for 5 Gbps link, and -11 dBm for a 10 Gbps link. This means a fiber optic transceiver could in principle support essentially any length link on a spacecraft, up to 400 meters at 10 Gbps if OM4 fiber is used.

Figure 4 illustrates the cable mass as a function of cable length.



Figure 4. Cable mass vs length. Note that electrical or optical connectors are not included.

	Low-Loss Coax Gore Type 21 Space-grade	SpaceFibre Coax (Axon AW2.2)	Fiber Optic 2 mm jacketed with 2 XCVR	
Cables required	4	4	2	
Loss	0.9 dB	6.4 dB	0 dB	
Cable mass (no connectors)	223 g/m	48 g/m	9.6 g/m	
Optical XCVRs	0	0	9.4 g	
3-meter total mass	670 g	144 g	38 g	
Power consumption	0	0	0.8 W	

Table 1. Summary of cable loss and mass at 10 GHz for a 3-meter link. Fiber optic cable mass (g/m) could be halved if a single jacket is used for both fibers, resulting in total mass of 28.6 g for the fiber optic link. Electrical and optical connectors not included in mass.

Table 1 summarizes the results of comparing the three implementations of the SpaceFibre physical layer. The muchlower fiber optic cable mass of 9.6 g/m, compared to 48 g/m for SpaceFiber Type A cable or 223 g/m for low-loss Gore Type-21 coax, results in greatly reduced overall physical layer mass compared to the coax solutions. The fiber optic cable mass could be essentially halved if a single jacket is used for both fibers, resulting in total mass of 28.6 g for the fiber optic link.

III. RADIATION TESTING OF SPACEFIBRE PHOTONIC TRANSCEIVERS

In a previous paper [2], we reviewed the design of photonic transceivers, which have two main sub-components: laser transmitter and photodiode receiver. The function of the transmitter is to convert electrical serial data bits to optical pulses, and the photodiode receiver converts optical pulses to

electrical serial data bits. These functions are realized in multigigabit systems using opto-electronic semiconductor devices (GaAs vertical cavity surface emitting laser (VCSEL) diodes and planar photodiodes at 850 nm) and SiGe BiCMOS electronic integrated circuit (IC) amplifier and control-loop devices.

Three form-factors were developed, including Size 8 optoelectronic contacts and printed-circuit-board mountable transceiver modules. All form-factors utilize the same circuit schematic and identical optoelectronic and electrical driver and controller devices. A wide range of environmental and reliability testing was performed on these devices and is summarized in [2]. These devices are shown in Figures 5-8.



Figure 5. Opto-Electronic Contact.



Figure 6. Size #8 contacts in panel-mount avionics connectors: D-sub (upper) and D38999 (lower).

The devices developed utilize a highly-simplified circuit approach with no microprocessor or memory devices in the units, so as to minimize susceptibility to radiation-induced single-event upset (SEU), single-event latch-up (SEL), etc. The Size #8 contacts and board-mountable transceivers were tested for resistance to radiation exposure in excess of 170 Krad total ionizing dose (TID) of gamma radiation from a cobalt-60 source, and 2.5 x 10^{12} neutrons/cm² of neutron irradiation. The units were operated with continuous data at 1.25Gbps and real-time bit-error monitoring, with no errors detected. Testing of the units conducted after the gamma and neutron exposures showed that the devices still met original specifications, and exhibited no appreciable degradation of performance.



Figure 7. PCB-mountable quad-output transmitter unit with ARINC 801 optical interface: Top view (upper) and bottom view (lower). Note: units are pictured without fiberoptic connectors attached.



Figure 8. Two-fiber PCB-mount transceiver form-factor. Top view with fiberoptic connectors attached (upper) and bottom view showing electrical board-mount high-speed connector interface at front edge (lower.)

Proton irradiation testing with 250 MeV protons from a synchrotron source was then conducted on the Size 8 contacts. Proton testing was performed at the Loma Linda University Medical Center (LLUMC) in conjunction with a third party. The testing was performed using the LLUMC synchrotron which provided periodic beam spills of 250 MeV protons incident on the Tx and Rx. The Tx and Rx were mounted on an evaluation card and oriented such that they were irradiated simultaneously during the testing.

During the testing a Bit Error Rate Tester (BERT) generates a 1.25 Gbps PRBS7 pattern that is input into the Rx under test and which then outputs to the Tx under test. The Tx finally outputs the pattern back to the BERT where it is compared to

the expected PRBS7 pattern. If any mis-compares are observed, they are logged. These mis-compares observed during irradiation are attributed to Single Event Upsets (SEU) resulting in data errors and contribute to the overall Bit Error Ratio (BER) calculated at the end of each run.

The observed SEU resulting in bit errors generally occurred during the beam spills and were typically isolated single bit error events, i.e. not burst events. During proton testing two radiation induced anomalies were observed resulting in a functional interrupt to the devices under test (DUTs). Additional analysis of these Single Event Functional Interrupt (SEFI) events is in process. The SEU events observed demonstrated little variance in their per-run fluence averaged cross-sections as can be seen in Figure 5. As only 2 SEFI events were observed, statistics are limited, but it is expected they would be rarely observed even in proton rich environments.



Figure 9: SEU and SEFI cross-sections for 250 MeV protons.



Figure 10: SEU Cross-sections as a Function of Accumulated TID.

During the testing parts were irradiated with 250 MeV protons resulting in an accumulated Total Ionizing Dose (TID) of ~450 krad(Si). A plot of the SEU cross-sections as a function of dose indicates negligible increase in SEU sensitivity as a function of accumulated TID as shown in Figure 10. The statistics on SEFIs were too low for evaluation. The SEU cross-section vs. accumulated TID results are plotted in Figure 10, overlaid by a linear-fit trend-line showing negligible change over accumulated TID.

Next, heavy ion radiation testing will be performed on the parts, and will be reported in future publications.

IV. ACTIVE OPTICAL CABLE TECHNICAL APPROACH

An Active Optical Cable contains one or more channels of optoelectronic transceiver functionality embedded in a cable assembly that has electrical connections for the high-speed inputs and outputs. All optical connectivity is embedded in the cable assembly. AOCs provide the benefits of high-speed fiber optic physical layer transmission, while removing the handling difficulties associated with delicate optical fiber interfaces, such as the need to inspect and clean optical contacts at each mating cycle. The functional block diagram of an AOC is illustrated in Figure 11.



Figure 11: Active Optical Cable (AOC) functional block diagram.

AOCs have been used extensively in datacenter networking applications, such as the QSFP+ 40 Gbps standard. However, ruggedized versions suitable for aerospace applications are not addressed by the commercial datacom components industry. A key difficulty has been the lack of a high-density, high-speed electrical interface connector solution that can withstand the rigors of the aerospace environment.

Due to the extremely high data rates up to 10 Gbps, the only contact types supported by aerospace-grade connectors capable of carrying these signals have until now been coax or twinax contacts that required termination directly to a coax or twinax cable. This limits the density of signals in a given size connector shell, and severely restricts the room available in backshell assemblies for AOC optoelectronics. These contacts also require time-consuming assembly techniques by skilled personnel.

To provide for higher density configurations and to ease assembly, we developed a high-speed electrical connector interface based on spring-loaded "pogo pin" contacts and pads inside of specially-designed high-density twinax contact arrays that can carry signals up to 30 Gbps. The pogo pins are terminated *en masse* to a flexible printed circuit board in a soldering process, eliminating the time-consuming handoperation of terminating individual coax or twinax contacts. This design also provides for seamless transition of the twinax contact to an impedance-controlled differential transmission lines on the flex circuit that can easily be interfaced to optoelectronic components or avionics circuitry [3], as shown in Figure 12.



Figure 12: Detail of high-speed "pogo-pin" connector external mating interface for D38999 AOC (upper); reverse view showing flex circuit attachment to the contacts (solder attachment of flex to contacts, and differential transmission lines not shown for clarity).

This rugged high-speed connector interface configuration can be utilized in various existing aerospace-grade connectors, such as D38999 and D-subminiature, micro-D, etc. The pogo-pin twinax contact construction permits higher density packaging than utilizing individual removable contacts, and also accurately preserves the impedance of the transmission line for superior signal integrity. Frequency response of 15 GHz and return loss measurements for this contact system are shown in Figure 13, implying data-carrying capability of up to ~30 Gbps.

We utilized this new high-speed electrical connector interface to enable a ruggedized AOC that employs the previouslydiscussed photonic transceivers in a backshell assembly. In the AOC plug, the flexible circuit board carries the high-speed signals to a transceiver contained in the backshell of the plug. Electrical power and monitor and control signals are also carried from the connector interface to the transceiver via a flexible circuit. Optical fibers connect the transceivers in the plug ends to each other. A complete AOC is illustrated in Figure 14 that uses D38999-type connector shells. Other connector types can also be supported.





Figure 13: Frequency response (upper) and return-loss (lower) of twinax contact pair showing -3dB bandwidth of ~15GHz.



Figure 14: Ruggedized Active Optical Cable (AOC) assembly illustration utilizing D38999 connector shells.

Figure 15 illustrates a cutaway view of the backshell showing the optical transceiver that converts the electrical high-speed signals to optical signals.



Figure 15: Cutaway-view of Ruggedized Active Optical Cable (AOC) plug showing photonic transceiver mounted inside backshell assembly. Optical fibers are not shown for clarity.

Figure 16 illustrates the receptacle to which the AOC plug is mated. The receptacle would be mounted to an avionics chassis to provide the interface for the AOC to the electronics inside the avionics box. The receptacle includes another high-speed flexible circuit assembly that carries the signals from the external connection interface of Figure 12 to the circuit board in the interior of the avionics chassis. A second low-speed connector is used to carry power, monitor and control signals.



Figure 16: Ruggedized Active Optical Cable (AOC) receptacle assembly illustration.



Figure 15: Detail of high-speed board-mount interface of Ruggedized Active Optical Cable (AOC).

The SpaceFibre standard also provides for EGSE interfaces using the commercial eSATA connector. Figure 16 illustrates another AOC that employs these interfaces for ground test applications. Figure 17 depicts the internal construction of the EGSE SpaceFibre AOC, showing the optical transceiver and optical fibers. Both of these AOC products are designed to meet the AOC requirements for the physical layer of the SpaceFibre standard.



Figure 13: EGSE version of SpaceFibre Active Optical Cable.



Figure 14: Internal construction of EGSE version of SpaceFibre Active Optical Cable.

V. PARALLEL OPTICAL TRANSCEIVER

Many benefits in size and mass can be obtained by moving from individual optical contacts, connectors and cabling to multiple-fiber "ribbon cable". A multi-fiber D38999-style connector utilizing a 12-fiber "MT" contact is illustrated in Figure 15. Such a connector solution is envisioned by the SpaceFibre standard, and requires suitably rugged parallel optical transceivers to efficiently make use of all of the fibers.

The SpaceFibre standard provides for multi-lane "parallel optical" transceivers in addition to single-lane devices, up to rates of 10 Gbps [1]. We developed rugged 10 Gbps parallel optical transceivers utilizing a hermetically-sealed hybrid optoelectronic microcircuit assembly and active optical alignment process, pictured in Figure 16. This construction provides for enhanced optical output power per lane of up to +3 dBm at 850nm and sensitivity of -12 dBm typical at 10 Gbps, leading to optical link budgets of up to 15 dB. The units incorporate an integrated heating element for the VCSEL laser arrays to permit high-performance operation over the temperature range of -40C to +85C, and can support data rates up to 14 Gbps per lane. 25 Gbps-per-lane parts are in development.



Figure 15: D38999 connector with a single MT contact (upper) and four MT contacts (lower).



Figure 16: Aerospace-grade parallel optical transceiver: top view with MTP optical cable inserted (upper left), bottom view showing high-speed electrical connector interface (upper right), transceiver mounted to evaluation board (lower).

The transceivers utilize an MTP optical connector that is specially modified to provide enhance vibration and shock survivability. The parts interface with the host board via a high-speed board-to-board connector with proven performance, identical to that used on the parts discussed in [1]. These parts have high reliability in excess of 1M hours at 50C based on FIT rate data for the lasers, photodiodes and other integrated circuits contained in the part.

They have been subjected to operating shock and vibration levels per MIL-STD-883 of 650 G 1 msec shock, 10 pulses in

each axis and 46 Grms random vibration, 2 hours per axis. The hermetically-sealed units also pass MIL-STD-883 helium leak rate testing as well as 10-day humidity testing with thermal cycling. For space applications, the removable finned heatsink pictured in Figure 16 can be replaced with a conduction-cooling heatsink that will reach to a conduction surface. The units were exposed to all of these tests while operating and bit errors were monitored at 5 Gbps. No errors were detected during any of these exposures. Radiation testing is planned on these devices, as well as extension of the bit rate to 25 Gbps, in the near future.

VI. CONCLUSIONS

Compact, rugged, opto-electronic transmitters, receivers, transceivers and active optical cable components were developed and tested for SpaceFibre physical layer

applications. Single and parallel transceivers were subjected to various tests, including thermal cycling, high vibration and shock, and gamma, neutron and proton radiation, and found to have excellent performance. Testing to date indicates that these parts can satisfy the environmental requirements of spaceflight applications. Further testing is ongoing, including heavy ion exposure, and results will be reported in future publications.

REFERENCES

- [1] ESA SpaceFibre draft standard, first issue, 2017.
- [2] R. T. Logan Jr., Spacewire Conference Proceedings, Yokohama, October 2016.
- [3] US Patent 9,664,868 R.T. Logan Jr., et al, "Advanced multigigabit connectors, inserts, active optical cables and methods," issued 5/30/2017.

Wednesday 16 May

Components (Short)

SPACE-QUALIFIED EUROPEAN LVDS Components for SpaceWire Networks

ARQUIMEA LVDS Family

Daniel González, Jesús López, Adrien Frouin Microelectronics Products and Services ARQUIMEA Ingeniería Madrid, Spain <u>dgonzalez@arquimea.com</u>, <u>jlopez@arquimea.com</u>, afrouin@arquimea.com

Abstract— Transmission of large amount of data is extensively used in communication among spacecraft and satellite onboard systems during most missions. LVDS (Low-Voltage Differential Signaling) Drivers and Receivers are key devices to provide means of sending/receiving data along twisted pair cables at very high data-rates with low power and excellent EMI performance. Rad-tolerant and Rad-hard ANSI EIA/TIA 644A complaint LVDS Driver and Receiver products are essential in an extensive range of space applications. Typical applications with such needs are SpaceWire and clock distribution networks.

The purpose of the paper is to present the main characteristics and features of the LVDS devices (i.e. repeater, driver, receiver and transceiver) developed by ARQUIMEA. They suppose a competitive alternative to the LVDS components currently used in many space equipment, especially those which are part of the SpareWire ecosystem, while providing higher speed, lower power consumption and extended input common mode range. The LVDS Repeater and Driver IPs have been characterized and evaluated according to ANSI and ESA standards. Qualification of the entire LVDS family manufactured in a single full-mask is in course.

Index Terms - SpaceWire, LVDS Networking, Spacecraft Electronics.

I. INTRODUCTION

The capabilities of remote-sensing instrumentation are developing rapidly. Consequently, the data rates being handled onboard spacecraft are increasing. Photo and video data transfers are a constant in many space missions. This pushes the need of connecting different systems with fast, reliable, low-power links and to achieve this, Low-Voltage Differential Signaling (LVDS) is a good candidate, since it addresses the requirements for high-performance data transmission applications. The LVDS standard is becoming the most popular differential data transmission protocol in the space industry.

LVDS delivers high data rates while consuming significantly less power than competing technologies. In addition, it brings many other benefits, which include:

Juergen Beister, David Levacq Electronic Components ESA - ESTEC Noordwijk, Netherlands Juergen.Beister@esa.int, david.levacq@esa.int

- Low-voltage power supply compatibility
- Low noise generation
- Good noise rejection
- Robust transmission signals
- Ability to be integrated into system-level ICs.

The LVDS technology allows communications to address high data rates in the range of hundreds of Mbps. For the reasons given above, the standard has been deployed wherever the need for high speed and low power exists.

LVDS is a differential, serial communication link, which uses low-voltage differential signals (350 mV_p typ.) generated from a current source that delivers 3.5mA on a 100 Ω termination resistor. It uses a typical common mode of 1.2V and limited slew rate to improve electromagnetic emissions. Point-to-point, single driver, single receiver and multipoint topologies, multiple drivers, and multiple receivers are compatible with the LVDS devices where tristate driver capability is present.

ARQUIMEA LVDS family composed by an Octal Repeater, a Dual Transceiver, a Quad Driver and a Quad Receiver offers these configuration capabilities with specific extended performances for space applications.



Figure 1. LVDS Quad Driver (Left) / Octal Repeater (Right)

II. MAIN CHARACTERISTICS

A. General

The LVDS devices were developed using a 0.25-um BiCMOS technology process from the German foundry IHP Microelectronics and observing special care during the design and verification phases to ensure full compliance with the ANSI_EIA/TIA644A standard, extended performance requisites, as well as reliability and radiation space requirements.

ARQUIMEA LVDS family offers very high performances. Over 500Mbps data transfer rate per channel is allowed over SpaceWire links. The devices provide 3.3V single power supply, low power consumption, small propagation delay, low channel-to-channel skew and low jitter.

Extended input common mode range from -4V to +5V allows compatibility between the power rails of the spacecraft embedded equipment.

Cold-spare functionality (essential for redundant systems architecture) is implemented in all pins, including CMOS Input/Outputs to allow high impedance state with very low sinking current when devices are powered off (VDD tied to VSS).

The input buffers of LVDS Receiver include an active internal fail-safe circuit that sets the output of the Receiver to a known high state when one or the two inputs are floating or shorted.

Input hysteresis implemented at the receiver side increases the noise tolerance.

The LVDS family is hardened against Total Ionizing Dose radiation above 300 kRad (Si), and is SEL immune up to LET of 60 MeV*cm²/mg. It allows a BER down to 1^{-13} err/bit in GEO orbit.

ESD protection above 8kV Human Body Model, +/- 250V for field induced Charge Device Mode and +/-500V Machine Model are implemented for all I/Os.

The most demanding requirements of the family (failsafe, extended input common mode and extended ESD protection) were driven by ESA. The development of the Octal Repeater was partly funded in the activity described in [1] and further presented and discussed in [2] and [3]. The development of the Driver, Receiver and Dual Transceiver devices as well as the complete qualification of the entire family was fully funded by ARQUIMEA.

B. Devices functionalities

The LVDS Octal Repeater (Figure 2) offers eight data channels plus an additional clock channel. Each channel is made of an LVDS Receiver and LVDS Driver. The Repeater has two dedicated Enable inputs for data and clock channels to set the LVDS outputs into tristate mode. It allows disabling the output stages and associated load current, and thus dropping the device to an ultra-low idle power state for multipoint topologies.



Figure 2. LVDS Octal repeater block diagram.

The Quad Driver (Figure 3) presents four data channels that convert LVTTL/LVCMOS inputs into LVDS signals. It implements a unique Enable signal to set all outputs into tristate mode.



Figure 3. LVDS quad driver block diagram.

The Quad Receiver (Figure 4) presents four data channels that convert LVDS inputs into LVCMOS output. It implements a unique Enable signal to set all outputs into tristate mode.



Figure 4. LVDS quad receiver block diagram.

The Dual Transceiver (Figure 5) implements a dual driver and a dual receiver with four independent Enable inputs to set separately each block into tristate mode.



Figure 5. LVDS Dual transceiver block diagram.

III. COMPARISON WITH THE STATE OF THE ART

A market comparison with available rad-hard components from Cobham and ST Microelectronics is presented in this chapter.

ARQUIMEA LVDS family is built on a fully European supply chain and is ITAR/EAR free. The LVDS family is Fit, Form and Function compatible with the existing products from the competition. Only the LVDS Octal Repeater for which a CQFP-48 package offers a competitive advantage in terms of body size (8x8 mm2) compared to the FP-48 (16x10mm2) Octal Repeater from Cobham or the FP-64 (21x8mm2) Cross-Point Switch from ST Microelectronics.

In general, all LVDS components are TTL compatible with cold-spare feature, qualified in the military temperature range and offer good performances against radiation (TID>300kRad, SEL free). Nevertheless, Cobham devices do not provide BER or SEU/SET results against particle hit on their datasheet.

ARQUIMEA devices allow higher data transfer rate (>500Mbps) than competition (400Mbps) with similar power consumption.

In relation to the Receiver design, ARQUIMEA devices allow extended input common mode from -4V to +5V (only 0V to +2.4V for Cobham) and high input hysteresis (apparently not offered by the competitors) while maintaining the input threshold range to ± 100 mV (± 130 mV for ST Microelectronics).

Concerning the Driver block, the nominal output current under 100Ω Load is around 3.5mA compared to the 10mA of LVDM devices.

8kV Full ESD protection is proposed by ARQUIMEA devices, when only 1.2kV is allowed by Cobham parts, and 8kV on LVDS outputs vs GND and 2kV on other pins is allowed by ST Microelectronics.

IV. EVALUATION RESULTS SUMMARY

In addition to radiation and characterization tests performed on the LVDS Driver and Repeater at extreme temperatures (whose overall results are presented in chapter II and III), an evaluation based on the ESCC2269000 standard was conducted to bring sufficient confidence for a successful further qualification of the LVDS family. These two devices are deemed representative of the LVDS family, since they implement the main blocks (Internal Regulator, Band Gap, LVDS Receiver/Driver, Digital Input pads, etc.).

Package selection from reliable manufacturers (as NTK and KYOCERA), die attach and bonding procedures (process, pad size, material, wire size and angle) were carefully settled and verified by the assembly house to reinforce the mechanical strength of the devices in harsh environment.

Absolute maximum rating and temperature stresses have shown good results on all tested devices, which gives confidence concerning their endurance for an extended period during space missions.

V. QUALIFICATION

The qualification of the LVDS family entirely follows the ESCC9000 standard and is being first conducted on the LVDS Quad Driver and LVDS Quad Receiver, according to their corresponding detailed specification. It includes production control, screening and Lot validation on representative samples of the family (Figure 6).



Figure 6. ESCC Qualification Flow

The Production Control considers Process Monitoring Review, Wafer inspection, Dicing and Assembly activities.

On-wafer electrical measurement at room temperature and die sort complete the production control to improve the yield.

The Screening considers assembly tests (i.e. thermal cycling, hermeticity, PIND, etc.) and a burn-in at representative maximum temperature and voltage ratings for 240 hours.

Since the automatic electrical measurement of all parameters is limited by the set-up performances, some parameters like skew time or jitter are proposed on a sample basis.

The Lot Acceptance Test consists of the validation of available data from former qualification and performance of Environmental, Mechanical, Endurance, Assembly Capability subgroup to complete the qualification (See Figure 7).



Figure 7. ESCC Qualification Tests

Appliance to ESA EPPL1 European Preferred Part List is foreseen at end of qualification activities by Q4-2018.

VI. QUALITY LEVELS PROCUREMENT

To offer a competitive lead time for the procurement of assembled devices and to reduce the lot validation tests effort, a significant number of packages is procured at once from a single lot. This ensures that all packages have the same construction and avoid possible concerns during the lot assembly capability verification.

The procurement options presented in the datasheet consider die format in trail and packaged devices in two quality levels: Engineering Models and Flight Models.

The die format in trail is procured directly through the assembly house, being the dicing process and post visual inspection performed according to space standards.

The assembled Flight Models, manufactured and validated according to ESCC9000 are available under a Minimum Order Quantity within illustrative 8 weeks lead time (depending on lot size).

The availability of Lower Quality Engineering Models with reduced screening and moderate verification (e.g. visual inspection, electrical validation at only 25°C) are available on demand under very lower order quantities within 6 weeks' lead time. EM samples can also be furnished with dedicated daughter boards for evaluation at system level.

A QML Q or V Delta-Qualification based on MIL-STD-38535 is available for Packaged Flight Models on-demand. For die Models, a delta-qualification based on higher level assembly specificities can be offered to comply with ECSS-Q-ST-60-05C or MIL-STD-38534 QML H or K.

References

- [1] ESTEC Contract 4000105886 with ARQUIMEA INGENIERÍA S.L.U
- [2] J. Lopez, E. Cordero, M. Cirillo, R. Dittrich, A. Frouin, R. Hansen and D. López "LVDS: A RAD-HARD Octal 500 Mbps Bus LVDS Repeater for Space," AMICSA 2016 proceedings.
- [3] A. Frouin and J. López "European LVDS Octal Repeater", TEC-ED & TEC-SW Final Presentation Days proceedings, December 2017

Comparative Performance of VITA 78 Connector Systems for Daughter Card to Backplane

Short Paper

Richard A. Johannes Smiths Interconnect Costa Mesa, USA richard.johannes@smithsinterconnect.com

Abstract— The transition from ground and aerospace to space frame applications, has driven an increase in the severity of the environment that the interconnect systems must support, while achieving solid signal integrity of high speed signals. These systems can no longer just satisfy the old military 1 µs maximum discontinuity signal interruption threshold requirement during exposure to these environments. Eye pattern monitoring with a mask or bit error rate testing is crucial to ensuring that signal integrity is maintained in a rigorous space application. In addition, by comparing the eye opening, it is possible to estimate the robustness of these systems in the form of a functional comparison at data rates up to 16 Gbps. This short paper explores the performance differences between two VITA 78 interconnect systems and discusses the impact of environment and system integration on connection performance. It also discusses the system considerations that change the actual performance of connection systems in contrast to the same connectors measured in isolation.

Relevant indexing terms: VITA 78, hyperboloid, ENIG, Wedge Lock, Signal Integrity, Jitter, Eye Pattern, Insertion Loss, Return Loss, and Characteristic Impedance.

I. INTRODUCTION

The VITA 78 specification was developed to provide state of the art ruggedized daughter card to backplane connection systems to the space community. There are two primary products that support this specification based on VITA 46 and 63. These interconnect systems support 6U, 3U and custom slot configurations to provide the necessary power, low speed, single ended high speed, and differential high speed signal transmission. The VITA 46 and 63 based systems are footprint compatible, but not intermateable.

This paper is intended to provide a general functional comparison between these products for global space applications. Contact technology, application and environmental compliance, press-fit or compliant pin performance, and differential high speed signal integrity will be discussed to provide a clear view of the relative capabilities.

II. CONTACT TECHNOLOGY COMPARISON

The connectors have distinctly different contact technologies. The first connector released under the VITA specification was designed around a card edge interface. This contact system started as a signal contact bump per beam, but due to poor vibration performance, was redesigned with two contact bumps per beam. While the additional contact points provide redundancy, there is not, as suggested in the literature, four true points of contact, because they are not supported on independent beams. Card edge contact technologies have been historically prone to PCB delamination during durability cycles or vibration induced movement. The pads on the PCB side of the interface have been extended and only minor plating edge degradation and some degree of gold wearthrough was observed during mate and unmate benchmark testing beyond 200 cycles.

VITA 63 saw the qualification of a second more robust contact technology, based on a space proven hyperboloid 360 degree multiple point contact. This contact technology has been the preferred method of meeting or exceeding the requirements of extreme mechanical shock and vibration.



Fig. 1 Card Edge Based Contact (VITA 46)



Fig. 2 Hyperboloid Based Contact (VITA 63)

The final contact consideration is that both connector systems have been tested for space applications with the primary terminations between the connector and daughter card and the connector and backplane based on eye of the needle press-fit or compliant pin terminations. The contact side is typically gold flash (< 20 μ in or 0.5 μ m) over nickel and the PCB plated through hole is ENIG (Electroless Nickel Plating with a thin layer of Immersion Gold). This interface has been shown to consistently meet the rigors of the space environment.

III. SYSTEM PACKAGING AND ENVIRONMENTAL Performance Comparison

VITA 46 and 63 are equivalent in terms of performance requirement with random vibration requirements to 11.7 Grms in the standard specification and 16.49 Grms in the more stringent specification. The greatest differentiation occurred during random vibration at 19.9 Grms and above with the VITA 63 having the strongest results. The vibration test was performed successfully on the VITA 63 connector to 25 Grms while monitoring Bit Error Rate (10⁻¹² was the target during shock and vibration). The Mechanical Shock testing was performed to 50 G (37 G for BERT monitoring) for mechanical integrity and 1 microsecond discontinuity.

The other consideration from a packaging perspective, involves the use of a wedge lock card guide for ensuring solid thermal contact between the daughter card and a cold plate. While the zero datum for the position of the back side of the daughter card should be at the point of contact with the cold plate, both connector systems are capable of absorbing a 0.016" (0.4 mm) lateral displacement.



Fig. 3 Wedge Lock Card Guide Assembly

IV. HIGH SPEED AND SIGNAL INTEGRITY PERFORMANCE COMPARISON

The signal integrity performance of connectors can be challenging to measure, given all the variables beyond the discrete connector that must be considered. Connectors, in isolation, can display characteristic impedances from 90 Ω to 110 Ω around a nominal of 100 Ω . If they are evaluated by themselves it would give the impression the connectors are equal in performance. This is not the case when system factors are added. If a connector with a characteristic impedance of 95 Ω is compared to one at 105 Ω , one might assume that the connectors will perform similarly. When integrated into a system, other contributing factors must be considered. The same 95 Ω impedance connector will drop to 80 to 85 Ω with the addition of the capacitance impact of the PCB plated through holes, while the 105 Ω impedance connector will measure as a 95 Ω impedance in system. The connector with the lower impedance will create a much larger disturbance in the signal integrity performance due to the greater impedance discontinuity from the 100 Ω nominal.

This example explains well the comparison between the VITA 46 and VITA 63 connectors, with the VITA 46 connector at about 95 Ω and the VITA 63 connector at 105 Ω . The images below highlight the electrical performance differences in detail.

A. TDR Impedance Performance in System (35 ps Edge Rate)



Fig. 4 VITA 46 – The Characteristic Impedance of the Connector in isolation is ~94 Ω with the minimum level reaching 83 Ω due to the capacitance to the PCB pin field at the backplane and daughter card.



Fig. 5 VITA 63 – The Characteristic Impedance of the Connector in isolation is ~103 Ω with the minimum level reaching 90 Ω , due to the capacitance to the PCB pin field at the backplane and daughter card.

The VITA 63 in Fig. 5, stays within impedance tolerance when added to the rest of the system, while the VITA 46 interconnect, in Fig. 4, drops well below the ± 10 % impedance tolerance when tested in system.

B. Insertion Loss Comparison







Fig. 7 VITA 63 Insertion Loss

In Fig. 6 the VITA 46 connector falls below the -3 dB specification in the 4GHz to 6GHz, while in Fig. 7 the VITA 63 connector maintains a uniform performance level throughout the frequency band providing solid insertion loss levels to about 16Gbps.

C. Return Loss Comparison



Fig. 8 VITA 46 Return Loss





As seen in Fig. 8 and 9, both connectors meet the -10dB specification for Return Loss, with the VITA 63 showing better return loss for data rates up to 20Gbps. The VITA 46 connector displays more harmonics or anomalies in the return loss trace, which will have an overall negative performance impact by comparison. This is potentially due to a less than optimum design of the internal print circuit boards used in the connector as well as the transition controls between the supporting contacts and the internal PCB's.
D. Eye Pattern Comparison



Fig. 10 VITA 46 Eye Pattern @ 16 Gbps

De-Embedded Measurements

			E J	Eye Wid itter: 19	dth: 29.9 9.14 ps	ps
15071C Heberek Analyzer						
tr4 sdd21, nigh = 400mV Low =	Ov. 15.9999994	1366bps				-81
450.0m 60.306738898/dps, -6.	671004566210ev					
400. 0m						
150.0m one: 136.4mv						
300.0m		10	1903			11
250. 0m						18
200.0		aitter: 9.961p				t: 130.1
150.0m						e1 gh1
100.0m						
10.00m			G103	14		
20101 46:41m				1		
0.000	and the second second		Hideb: 45 03er		Statement of the statement of the	
-50.00m	0	15.625001p	11.250001p	46.8750020	62,5000028	91,750001318756
1 Start 12 091895 104:		SFB// 70 kHz		Stat	19.999999292 GH2	(Pint) CT THE HARMON

Fig. 11 VITA 63 Eye Pattern @ 16 Gbps

De-Embedded Measurements

Eye Height: 130.3 mV Eye Width: 45.03 ps Jitter: 9.961 ps

Eye Height: 78.27 mV

The VITA 63 connector pattern in Fig. 11, shows a 40% greater eye opening height as compared to the VITA 46 connector. The VITA 46 system in Fig. 10, displays a great

deal more noise, with a staggering 92% higher level of jitter. Based on the eye patterns above, the VITA 63 should be capable of supporting 16 Gbps data rates assuming typical noise levels in the balance of the system transmission path. The VITA 46 connector is not able to support this data rate and is limited to about the 10 Gbps signal transmission level based on the performance test data.

V. CONCLUSIONS

The information presented in the above sections clearly shows that the hyperboloid based VITA 63 interconnect system outperforms the PCB (card edge) design approach use by the VITA 46 connectors. The primary areas in which this differentiation has been highlighted in the benchmark testing are in the mechanical shock and vibration operational tests and in the signal integrity testing (particularly above 10 Gbps).

It must also be noted that both systems use eye of the needle press-fit or compliant contact technology for interface to the daughter card and backplane with solid environmental test performance. These interconnect systems meet the minimum requirements of VITA 78 and space performance.

The VITA 63 interconnect will provide superior performance to not only meet the current space application needs, but to support future increases in mechanical stress, reliability and higher data rate signal integrity demands for daughter card to backplane architectures.

REFERENCES

- [1] Anritsu, "Application Note No. 11410-00533 Rev. A", Anritsu Company, 2010
- [2] ANSI/VITA (2015), ANSI/VITA 46.0-2015 VPX Base Standard
- [3] ANSI/VITA (2015), ANSI/VITA 46.11-2015 System Management on VPX
- [4] ANSI/VITA (2015), ANSI/VITA 63.0-2015 Hyperboloid Alternative Connector for VPX
- [5] ANSI/VITA (2015), ANSI/VITA 78.0-2015 SpaceVPX System
- [6] D. J. McKenney, M. Shorey, Vita 72 VPX Connector Update for Open Standard Module Vibration Reliability Testing Rev 1.1, Mercury Computer Systems, 2012
- [7] M. R. McAlonis, "Expanding a Strong Foundation: "Ruggedizing" Connector for Next-Generation Embedded Computing Applications, Tyco Electronics Corporation, 2013

Modular Interconnect for Point to Point and Backplane Space Application Short Paper

Richard A. Johannes Smiths Interconnect Costa Mesa, USA richard.johannes@smithsinterconnect.com

Abstract— Space architectures are continuing to follow the trend of either daughter card to backplane or point to point system designs. The highest degree of flexibility is typically achieved using a point to point approach, as seen in frame slice and Space Age Bus architectures. There is an ever increasing demand for rugged, high speed (up to 10 Gbps) modular and configurable connections, with the combination of controlled impedance differential signal pairs, fiber optic pairs, power pins, and discrete pins in a single connector housing. This technology will revolutionize the way in which space craft are designed and built. The satellite market is moving away from RF Analog based payloads providing low speed telecommunication signaling, to a new Digital Transparent Processor architecture for high throughput satellites. Smiths Interconnect has developed, tested and qualified such a connector. This connector is an advanced high speed interconnect solution to offer next generation data on demand, meeting both point to point and backplane connector requirements. The system has proven to be able to withstand space application requirements, including extreme levels of vibration, shock and climatic testing and offers a reliable way to implement high density interconnections with high speed signal transmission requirements.

Relevant indexing terms: attenuation, twinax, quadrax, MT Fiber Optic ferrule, jitter, return loss, insertion loss, Space Age Bus, backplanes, and electromagnetic interference

I. INTRODUCTION

There is an ever increasing demand for rugged, high speed (> 6.25 GBPS), modular and configurable connections, with the capability to support controlled impedance differential signal pairs, power pins, and discrete pins in a single connector housing. The satellite market is moving away from lower speed RF analog based payloads providing for telecommunication signaling, to a new Digital Transparent Processor architecture. This technology was introduced and qualified in 2017 for 10 Gbps PCB to cable applications. Smiths Interconnect has developed, and tested this interconnect system for specific applications. Smiths is now finalizing the launch of the full platform of modules to support advanced high speed interconnect solution to offer next generation data on demand, power, fiber optics and low speed signal capabilities to meeting both point to point and backplane connector requirements. The system has proven to be able to withstand space application requirements, including extreme levels of vibration, shock and climatic testing and offer a reliable way to implement high density interconnections with high speed signal transmission requirements.

II. NEXT GENERATION INTERCONNECTION SYSTEM

This connector series has been designed for extreme levels of shock and vibration with a new micro-boloid contact technology, and the option of aluminum or lower mass using innovative composite shell materials with gold over nickel, and nickel plated variants. The system has also been updated and is expected to exceed electromagnetic interference shielding requirements of D38999 (the target is -70 dB attenuation at 10 GHz). It is designed with housings to support 4, and 8 bay configurations. These bays can accept any combination of 7 highly configurable and interchangeable modules. It is blind mateable, hot pluggable (for voltages <5V), and scope proof, with ultra-low mating forces and low outgassing materials. The connector features a very low mass composite material shell, translating into a lightweight, yet robust construction, well suited for extreme levels of shock and vibration near the launcher.



Fig. 1 Basic Modules



Fig. 2 Fiber Optic Modules

III. MODULE CONFIGURATIONS (FIG. 1 & 2)

- a. Split Quadrax Two Twinax Cables in a Common Contact Assembly (100 Ω each pair)
- b. Twinax Discrete Twinax with Two / Bay (100 Ω)
- c. Power Module -4×5 A Contacts
- d. Mid-power Module 5 x 3 A Contacts
- e. Low Speed Signal 10 x 1.5A contacts
- f. MT Fiber Optic Ferrule up to a 12 fiber single mode ribbon
- g. MT Fiber Optic Ferrule up to 12 fiber multi-mode ribbon

Additional modules are under development including a 50Ω coaxial contact, as well as PCB mount configurations. At present, only a. and b. above are fully qualified with the balance of the module qualifications in process.

IV. ELECTROMAGNETIC INTERFERENCE PROTECTION

The EMI protection level is targeted at -65 dB attenuation for the carbon filled composite housing systems and -70 dB attenuation for the aluminum housing systems, measured at 10 GHz. The EMI shielding is provided by an encapsulating backshell on the plug side of the interface, which engages an EMI grounding spring on the receptacle side housing at the bulkhead. This allows the shield on the exiting cable to be terminated on the contact outer body for the twinax contacts and at the backshell exit for the other copper conductors. The fiber optic cables will not need to be shielded beyond the exit of the backshell, with any energy radiating from the rear of the connector attenuated within the backshell cavity (Fig. 3-6).



Fig. 3 EMI Backshell and Ground Spring



Fig. 4 Bulkhead Mounted In-line Configuration



Fig. 5 Bulkhead Mounted PCB to Plug Configuration



Fig. 6 Plug Assembly Encapsulated in EMI Backshell

V. SPLIT QUADRAX AND DISCRETE TWINAX QUALIFICATION

The high speed modules have been developed in two configurations. The first configuration has two twinax cables combined into a single, four inner terminal, quadrax contact (Fig. 7). The second configuration has two discrete differential twinax cables terminated to their own discrete twinax contact,

keeping the grounding outer body separated by differential contact pairs (Fig. 8).



Fig. 7 – Split Quadrax (Combined Twinax)



Fig. 8 - Discrete Twinax Contacts

The original high speed module housing was constructed and qualified using gold plated conductive PEEK plastic to provide basic ground continuity and EMI shielding. The next generation version of the product (currently undergoing qualification), has two housing material options for improved performance. These options are gold plated aluminum and gold plated conductive liquid crystal polymer. The new materials will provide superior ground continuity and in the case of the LCP, significantly enhanced moldability. The newly integrated EMI encapsulating backshell will also provide a very high degree of EMI attenuation.

Both the existing and new connector designs share the same stable controlled characteristic impedance as seen in Fig. 9. The impedance of the connector is maintained at 100 $\Omega \pm 5\%$. The termination to the PCB is accomplished using a spring probe interface, thus eliminating the effects of a large plated through-hole for a solder tail or a press-fit pin. The time domain reflectometer (TDR) impedance results will be further improved, with either an optimized PCB or cable to cable in-line connection. The non-optimized PCB can be made more electrically consistent through control of the depth and diameter of the blind vias used to connect the surface target pads to the internal traces. This would significantly reduce the capacitive dips seen at the PCB to connector transition.



Fig. 9 - End to End TDR Trace - Non-optimized Test PCB

The high speed performance observed during qualification, met all requirements before and after environmental conditioning. The readings were completed using a non-optimized PCB and bent twinax cables, to further evaluate end to end worst case conditions. The eye patterns shown below represent system performance at 3.75 Gbps (Fig. 10), 6.25 Gbps (Fig. 11) and 10 Gbps (Fig. 12). The system results in the bent cable condition were required to meet a jitter level of 96 ps max., differential skew of 35 ps max., Insertion Loss (7dB max. @ 5 GHz & 15 dB max. @ 10 GHz) and Return Loss (20 dB min. @ 10 GHz), after thermal aging, thermal cycling, mechanical shock and vibration.



Fig. 10 – Eye Pattern at 3.75 Gbps (Jitter 25.83 ps)



Fig. 11 – Eye Pattern at 6.25 Gbps (Jitter 29.5 ps)



Fig. 12 – Eye Pattern at 10 Gbps (Jitter 40.63 ps)



Fig. 13 - High Speed Test Set-up - Agilent Network Analyzer

The connector was tested electrically before and after conditioning under vibration (X & Y-axis to 38.1g rms and Z-axis to 41.4g rms), mechanical shock to over 2,000g's and thermal cycling (-55°C to +125°C for 110 cycles).

VI. CONCLUSIONS

The ELARA modular connector systems has demonstrated very solid performance in both ECSS based testing as well as the additional application specific high speed signal integrity evaluations. The connector system has already proven its ability to meet the next generation of flexible interconnect application, as demanded by DTP and SpaceWire. The connector has completed qualification for the high speed modules for space applications. The platform qualification and release of the interconnect system is underway and is intended to augment the existing product with a backwards compatible interface. This release will include greater capability in EMI attenuation, as well as the completion of power, low speed signal and MT ferrule based fiber optic modules. In the future, additional modules are planned to support other fiber optic variants, higher density and 50Ω coaxial interconnects.



Fig. 14 – 8 Bay Plug Assembly



Fig. 15-4 Bay Connector Assembly

REFERENCES

- R. Johannes, M. Carlson, "A Highly Configurable Modular Backplane Connector for Point to Point and Backplane Connections Suited for Space Wire, Space Age Bus, and Frame Slice Architecture", Smiths Connectors, 2016.
- [2] S. Honour, Qualification Test Report Connector Assembly Nexus 8 Bay (8 X High Speed Quadrax Modules) with Twinax Loop Cables, Revision A.1, 2018.

Control Loop Processor: A reliable and agile processing platform for mission critical applications

Short paper – Components session

RUIZ Marco TEC-EXP department SABCA Chaussée de Haecht, 1470 1130 Brussels, Belgium marco.ruiz@sabca.be

Abstract - SABCA has developed with ESA support the Control Loop Processor (CLP), a new specific processor under integration in next generation European space vehicles (Vega-C, Ariane 6, Space Rider). The determinism of the processor architecture and associated embedded software ensures the complete system behaviour predictability, a key feature for tomorrow intelligent systems in charge of mission critical processing (space exploration robot motion systems, space science mission telecommunication systems, human space exploration life systems, etc). Such approach can be of interest of future Spacewire applications which are becoming more challenging and therefore may impact the performance of control loops running on a node.

This paper aims at explaining how data transfers over a SpaceWire link can be performed on a deterministic processing platform such as the CLP. The specificities of the Control Loop Processor's architecture and application development environment will be first presented. The software architecture to keep the solution determinism while performing asynchronous data transfers will then be given with a few representative operational examples.

Keywords—Determinism, processor, reliable, agile, platform, SpaceWire, mission critical

I. INTRODUCTION

Nowadays, the field of tight control loops, characterized by hard real-time constraints (loop frequency > 1 kHz) in conjunction with complex algorithmic needs, currently lacks a microprocessor allowing to make a software programmable approach economically and technically viable. The control of electro-mechanical actuators is an example of target application.

A dedicated space-hardened microprocessor, called Control Loop Processor (CLP), has been developed - with the tight collaboration of **ESA TEC/ED** - and integrates several key features ensuring a fully deterministic behaviour as well as an embedded robustness/anomaly management in conjunction with vectorial IEEE-754 floating-points operations. FERON Jean-Brieuc TEC-DME department SABCA SABCA Chaussée de Haecht, 1470 1130 Brussels, Belgium jean-brieuc.feron@sabca.be

The Control Loop Processor has the unique specificity of being a fully deterministic soft processor, i.e. having a completely predictable behaviour. This predictability provides a guarantee of its processing performance in operation. Consequently, it targets applications requiring flexible processing with the highest certified Quality of Service (QoS). The Control Loop Processor offers a guarantee of robustness and reliability to an application. It is the last member of a family of flight-proven soft & hard processors with more than 20 years of R&T legacy and is under integration in next generation aerospace vehicles.

It is available for any application as a low cost softcore without any export restriction. It comes with a complete development platform and competitive support, customization and integration services. Additional activities are foreseen to consolidate the CLP ecosystem

II. THE CONTROL LOOP PROCESSOR

The Control Loop Processor is made of **two CPU**s and a number of **peripherals**. Its functional diagram is provided in Figure 1



Figure 1: Control Loop Processor functional diagram

Each CPU is based on a Reduced Instruction Set Computer (RISC) architecture and cache-free topology to provide a fully deterministic behavior with no interruptions. It is made of three Arithmetic Logical Units (ALUs): two are Floating-Point Units (FPUs) and the third one is a 16-bit integer unit (called SU). The floating-point units use the IEEE-754 format in single precision but can also use 16-bit integer representation. Various Single Instruction Multiple Data (SIMD) instructions are available to parallelize computations.

Each CPU contains a 32kx39 **program memory** (PRAM), which is also available for data through an AMBA Peripheral Bus (APB) controlled by a single CPU. Each CPU integrates 512 general-purpose registers in each floating-point unit and 32 general-purpose registers in the integer unit. A stack pointer is also available with an on-chip integration of stacks (using general-purpose registers). Each CPU supports **Direct Memory Accesses** (DMA), which allows to off-load the software from transfers occurring between the CPU registers and any APB peripheral. The communication between the two CPUs is possible through a 1024x39-bit wide Intercom RAM (32-bit for data and 7 bit for EDAC).

Two on-chip busses, based on the **APB** standard, are available. Each CPU has its own bus to interact with peripherals and has the master role. Each peripheral is thus exclusively slave. An allocation and write protection mechanism is integrated to program, at boot time, which APB register or segment is allocated to which CPU. When the software is running, this configuration is static thus ensuring a deterministic communication between the CPUs and the peripherals. Any abnormal access is detected and reported in a dedicated counter.

The ten on-chip **Timers** are available for use by the CPUs or most of the various on-chip peripherals. These timers are central to the CLP as they allow synchronizing the software running of the CPUs with the used peripherals.

Spacewire and Spacewire-RMAP interfaces (as well as MIL-STD-1553 RT and CAN interfaces) are included to support most space applications needs. All these interfaces comply with their respective standards and interact with the software through two eXCHange RAMs (XCHG RAM). Each CPU has its dedicated XCHG RAM which is 8Kx39 bits wide. Section III provides a more detailed presentation of this interface.

A **Memory Management Unit** (MMU) performs the routing from/to these XCHG RAMs from/to the mentioned interfaces. The MMU can be programmed by the software through descriptors. An allocation mechanism is also foreseen to filter data handled by a given interface.

. The CLP memory space is **fault-tolerant**. If the right silicon technology is selected, it can be SEU (Single Event Upset)-tolerant. EDAC (Error Detection And Correction)

management is integrated to perform "on-the-fly" correction in case of read and an on-chip write-back correction to avoid errors accumulation. A scrubbing unit is also foreseen with a programmable refresh rate and correction address range. The fault tolerance also applies to any abnormal event occurring inside the CLP such as a deadlock condition in the ADC interface, an allocation violation detected by the MMU, a saturation condition occurring in an arithmetic unit or an illegal operand detection.

The software is debugged and validated through the Real-Time Background Tracer (RTBT) link. This interface is based on the 100 Mbits/s Ethernet/UDP standard and allows to perform real-time tracing of CPUs registers, APBs traffic, MMU interactions and code execution. The RTBT is nonintrusive and does not need instrumentation code thus making it suitable for applications forbidding useless code

In addition, four UARTs are included to interface any off-chip device with the CLP. The baud rate is programmable and either 8-bit or 32-bit modes are available. The CLP also includes one I^2C interface to interact with devices using this standard. Two CRC units are available to perform data integrity check. The polynomial is fully 96 GPIOs are available either in the programmable. conventional mode (input, output or input/outputs) or in a MEM8 mode allowing to automatically interact with an external 8-bit memory. These GPIOs are multiplexed with other interfaces in the IOMUX pins. Two SPI interfaces are included and can be programmed in either master or slave mode. The data acquisition can be either sequential or parallel. This interface is particularly interesting when working with serial off-chip ADCs. A Pulse Width Modulation (PWM) interface is integrated allowing to drive up to 2x24 signals. Each output has its complement thus easing the generation of signals intended for power electronics. Two Arbitrary Waveform Generators (AWG) are available to automatically generate any arbitrary waveform. The typical use of these AWG is either performing sensors excitation or analog reporting. A parallel ADC interface is included to allow interacting with any off-chip parallel interface. The polarity and timings are fully programmable. Up to 4 off-chip devices can be addressed and 16 samples transferred per acquisition (which is triggered by one of the CLP timers). The ADC interface is autonomous.

Finally, a boot manager is included which provides a robust boot and restart mechanism through four boot descriptors. Five boot sources can be programmed (RMAP through the primary or redundant SpaceWire interface, CAN, MEM8 or RTBT) to automatically upload these boot descriptors and then boot the whole CLP according to the content of these ones.

III. SPACEWIRE INTERFACE ON THE CLP

The CLP has two CLP SpaceWire/RMAP interfaces supporting both the SpaceWire messages, according to ECSS-

E-ST-50-12C, and the RMAP messages, according to ECSS-E-ST-50-52C. In particular, the CLP supports the SpaceWire LVDS encode-decoder specification, the RMAP target only specification, the SpaceWire RMAP Write specification, the SpaceWire RMAP Read specification, the SpaceWire RMAP Read-Modify-Write specification and the SpaceWire system time distribution.

One CLP SpaceWire interface allows to handle up to 64 different transmit messages and up to 64 different receive messages .This is achieved by defining a transmit descriptors table and a receive descriptors table that are respectively made of 256 words and 128 words. These tables are programmed in the XCHGRAMs. Each transmit descriptor is made of 4 consecutive words. Each receive descriptor is made of 2 consecutive words.

Each table works as a circular buffer. This means that when a message has been transmitted, based on the content of a given descriptor, the SpaceWire interface will automatically send the message associated to the next descriptor. The same occurs for a message that is being received. In that case, this one is processed according to the current receive descriptor. When the transfer is ended, it automatically process a new incoming message with the next descriptor of the receive descriptor table. This implies that the **maximal time** needed to process an incoming frame or to transmit an outgoing frame from/to the XCHG_RAM can be determined for a given application layout. The transmit and receive tables location and the value of the current descriptor are maintained in APB registers. Note that any descriptor can be enabled/disabled no matter what the value of the current descriptor is.

The Memory Management Unit (MMU) works in tight interaction the two SpaceWire interfaces (cfr Figure 2). It should not be confounded with the classical virtual memory handling. It is rather a sort of intelligent router that is configured by software and allows a background management of these interfaces to/from the two XCHGRAM(s) through a DMA mechanism (not to be confounded with the one in the CPU). This feature is crucial for the CLP predictable architecture as it allows to **off-load the software** from lowlevel tasks related to complex interfaces such as the SpaceWire. The figure below shows a simplified view of the MMU, XCHG_RAMs and CPUs system (the Spacewire is on the right and connected to the MMU).

A number of protection mechanisms are also available to detect any event occurring between the MMU and the peripherals or to trace an address corruption due to a misconfiguration.

The permanent reception or transmission of Spacewire messages working up to 200 Mbit/s is supported. This figure includes the latency needed by the data/control flow to go from the Spacewire I/O interfaces up to the corresponding XCHG RAMs (and vice-versa). Note that this figure does not

take account of the speed at which the data is retrieved or posted by the software.



Figure 2: MMU and CPU interactions

IV. DETERMINISTIC APPROACH IN NON-DETERMINISTIC INTERFACES

An application running on the CLP is constructed with **timeslots** to fully take advantage of its predictable architecture. A number of predefined and fixed tasks are executed within each slot. It is up to the user to decide the number and duration of each time slot. Each duration has to be associated to a given time budget which is predetermined during the architectural software design.

Each time slot length is controlled by software with one of the available CLP on-chip timers whose tick triggers the start of execution of the allocated tasks. When these ones have been completed, the software is halted until a new synchronisation tick is detected. This event resumes the software execution which starts the next task(s) of the following timing slot. There is no technical limitation on how the time slots, timers and CPUs must be used. Their configurations are entirely left to the user. In some cases, it can be decided to let faster loops be executed on one CPU while higher level tasks can be performed on the other CPU. In some cases, more symmetrical behaviours can be preferred to have a better balance of tasks. Various architectural layouts can be assessed to benchmark their impacts in terms of performance to select the best time slots topology.

The ticks produced by the on-chip timers may also be detected by a number of others interfaces (e.g. PWM to produce synchronous commands) to synchronise specific event such as frame transmissions or ADC acquisition. This provides to the software the full control over the various interfaces activity thus greatly facilitating its design.

The Figure 3 provides an example of an application using one CPU to implement three control loops at various

frequencies. Each tick signal (Sync in the figure) also triggers specific interface events.



Figure 3: Example of architecture based on time-slots

Because of the inherent cycle-level predictability, the user is always guaranteed that a task is always completed within the allocated time. This feature is useful for control loops which need to be executed at the right frequency to sustain the application performance. However, another implicit advantage of this approach relies on the fact that the handling of a communication interface - such as the SpaceWire - is made easier.

Indeed, as shortly presented in section III, the presence of a dedicated XCHG RAM dual-port memory allows to off-load the CPU from interruptible tasks which may degrade the performance of control loops. Each CPU has its dedicated XCHG RAM memory which also physically prevents buffer overwriting effects. Basically, the only task which is still under the software control consists in regularly checking the descriptors contents (programmed one during the initialisation) by polling the right status bits/words, process data which has arrived and post new data which needs to be transmitted. Note that underrun and overrun conditions can also be detected via dedicated status flags.

When a Spacewire application implements these communication principles via one or several tasks executed in time slots, the user has full control over the **maximal latency** needed by the software to handle Spacewire frames maintained in the XCHG_RAMs. As stated in section III, since the maximal time needed to transmit or receive a Spacewire frame from/to the XCHGRAMs can be determined, a robust exchange mechanism can be setup ensuring on-time data delivery and processing while **minimizing the software overhead** at the level of control loops runtime.

The validation of this implementation is also made easier because the worst-case execution time – whose calculation is mandatory for safety-critical applications - can be naturally determined by analysis and later tested on the hardware (via the RTBT link presented in section II). This simplifies the engineering work and therefore should significantly reduce the application non-recurrent cost.

It is believed that the approach presented in this paper is suitable for applications needing light to moderate Spacewire interactions. Another advantage may be a cheaper solution since the use of a dedicated Spacewire chip is avoided thus decreasing board space and cost. For higher bandwidth Spacewire applications (such as routers), the user may still need the use of a dedicated chip.

This approach is not limited to the Spacewire and applies to each CLP interface (including simpler ones such as the UART and SPI or more complex ones such as the MIL-STD-1553. It has been extensively used on various applications such as the Thrust Vector Control of the 4 stages of the Vega Launcher (based on the MIL-STD-1553). Another use case is the TVAS of the Ariane 6 launcher which is based on 100Mbit Timetriggered Ethernet (whose protocol aims at being deterministic at the network level as per the Spacewire-D).

V. THE SOFTWARE DEVELOPMENT ENVIRONEMENT



Figure 4: Software Development Environment

The **SDE user interface** is based on two standard tools which are commonly used in space applications:

- Matlab/Simulink to graphically design the application with built-in entry methods and perform simulation
- Eclipse which is an open-source framework giving a user-friendly interface to all the SDE items that will be developed

The SDE generator is intended to produce an intermediate level model from the high-level model that has been designed by the front-end tool. The generated model is compliant with format defined by LLVM library and can thus be used as an entry by the SDE compiler. The SDE generator will make use of the off-the-shelf Qgen tool, provided by ADACORE company.

The **SDE compiler** is intended to produce a target independent low-level model from a model described with C standard. It can also read an intermediate-level model which is compliant with LLVM format. The generated code can be as an entry for the SDE assembler. No license or fee is needed to access this library The **SDE assembler** is intended to produce a binary code from a low-level representation model. A first prototype has been designed for this tool and is available on request. It is intended to update and enhance the development with a number of additional utilities such as a configurator allowing a user to initialise the CLP peripherals and CPU(s) in a friendly fashion, a loader allowing the user to download the binary code into the CLP board, an assembly editor based on Eclipse

The **SDE debugging tool** is intended to support the user to test, debug and validate an application running on the CLP. Four items are foreseen: the unit test manager giving a basic infrastructure to the user to develop unit test campaigns on CLP assembly code, the tracer allowing to program and track all reporting information to/from the RTBT unit inside the CLP, the ISS based on the cycle-accurate CLP SystemC model, the debugger allowing the user to interactively debug a given CLP code by means of various intrusive functions. The debugger will make use of the SystemC model whose behaviour is ensured to be equivalent to the CLP because of the co-validation and co-simulations activities performed during the validation.

When a CLP application is designed by mean of the Matlab-Simulink tool, two different approaches are available. The user may use the built-in standard library (i.e. S-function, and so on...) or a dedicated library made of custom blocks coupled to CLP assembler code or a mix of both approaches. To support the user in this second alternative, a **library** with a number or predefined **CLP macros** including commonly used high-level functions have been developed. These ones cover, for instance, complex mathematical functions such as filters and low-level drivers functions allowing to simplify the communication with the CLP interfaces (SpaceWire, PWM, MIL-STD-1553, etc...).

VI. VALIDATION CAMPAIGN

Eight different testbenches have been developed to extensively verify the CLP RTL code through simulations and FPGA prototyping.



Figure 5: Testbenches architecture

The validation methodology is given in

and aims to generate a number of stimuli which are applied to a reference model and the DUT in an automated fashion. The generated data are then compared and a report is created to provide the pass/fail status and reproduce the problem if required.

For most CLP functions, the reference model is a **cycleaccurate SystemC model** that has been developed in parallel with the VHDL code.

The SystemC model is foreseen to be integrated in the ISS as well as the various SDE tools such as the debugger of the unit test manager to take advantage of its full cycle-level accuracy. Stimuli are created by a tool generating directed stimuli and /or pseudo-random vectors according to a predefined test scenario. The stimuli then feeds the DUT and its corresponding reference model. The DUT may either be the CLP RTL code or an FPGA prototype – based on the RASTA system from Cobham Gaisler (cfr *Figure 6*) - integrating the full CLP design in a Virtex-4 device.



Figure 6: RASTA system with CLP integrated

More than 600 hundreds test campaigns based on Perl and Python scripts have been developed to extensively cover the CLP design and ensure the maturity level requested by ECSS standard.

VII. CONCLUSION

The Control Loop Processor architecture aims to bring a technology initially dedicated to high performance motor control in harsh environment to new applications and new markets. This paper presented to main features of the CLP and how its determinism may help a user to simplify the management of a Spacewire communication link, which is inherently non-deterministic

Today, the Control Loop Processor is available as a softcore microprocessor, i.e. it can be implemented in any

suitable silicon support (ASIC, FPGA, etc.), offering a wide range of performance in terms of clock speed, power consumption and area footprint. To offer maximal flexibility in terms of use, this processor architecture is highly customizable: the number of cores, the number of arithmetic units, the memory size, etc. can easily be modified and automatically re-validated. Special attention was given to ease the use, integration and software development effort of this solution. The result is a highly reliable and robust deterministic processor coupled to an ergonomic development platform Future activities are under preparation regarding a radiation-tolerant ASIC development of the Control Loop Processor.

ACKNOWLEDGMENT

We would like to thank the European Space Agency and in particular Roland Weigand, our technical officer from ESTEC's TEC/ED department), that supported this activity from its beginning and provided precious advises during the development.

Galvanic Isolation of SpaceWire Ports – An Innovative Design Approach

SpaceWire Components, Short Paper

G. Baterina ICE-O Tech Las Vegas, United States of America gil@ice-o.tech

Abstract-This paper reviews the summary of needs for galvanic isolation in SpaceWire networks and the limitations of current isolation solutions; the paper also proposes a new approach to achieving galvanic isolation of SpaceWire ports through the use of commercial off-the-shelf (COTS) components. This innovative design approach overcomes the non-DCbalanced nature of SpW signaling and enables the use of readilyavailable conventional capacitive (capacitor-based) and/or inductive (transformer-based) isolation barriers; to simplify implementation, adaption, and integration, the design approach does not require any modifications to the existing SpW protocol or network structure. To demonstrate this solution, a discrete SpW galvanic isolation receiver module will be built utilizing COTS components such as LVDS, CML, and LVCMOS transmitters and receivers, logic devices, capacitors, and/or transformers; although the demonstration module will only be isolating the receiver lines of an SpW port, if needed, the symmetrical design will allow the utilization of a duplicate module for isolating the transmitter lines as well. The proposed integrated module is expected to have on-board isolated power and operate up to 400 Mbps, handling a common-mode voltage of 100 V-RMS, and a galvanic isolation of 1 kV-RMS.

Index Terms— SpaceWire, SpW, isolation, fault propagation, LVDS, common mode voltage, galvanic, component, spacecraft electronics, commercial-off-the-shelf, COTS.

I. INTRODUCTION

SpaceWire (SpW) as defined in the standard [1] uses the Low Voltage Differential Signaling (LVDS) electrical interface which has the advantage of reducing the power required for a high speed data link; however, the existing LVDS buffers and ASIC devices have 2 principal drawbacks for implementing high reliability systems:

- limited common mode voltage tolerance
- fault propagation paths

The common mode tolerance is +/-1V; if this voltage is exceeded then the link data may be corrupted. In the worst case the transmitter/receiver devices may either be stressed or permanently damaged. Stressing of the LVDS buffer may not be evident but often results in a reduced reliability leading to

A. Senior Thales Alenia Space UK Ltd Bristol, United Kingdom alan.senior@thalesaleniaspace.com

premature failure later. Within a spacecraft, it is practical to control the common mode voltages within the specified limits and thus once launched problems would not be anticipated. Control of the common mode voltages during ground testing of spacecraft with remote Electrical Ground Support Equipment (EGSE) that use long cables becomes more problematic; drivers and receivers have failed in test configurations either due to incorrect test setups, poor grounding setups, or the effects of EMC testing. Clearly it is important to implement an effective grounding scheme and ensure that methods for monitoring the common mode voltages are in place rather than wait for failures to occur or assume acceptable conditions are met.

Fault propagation paths exist between LVDS link ends due to the direct silicon to silicon connection between the devices at the two ends of a link [2]. As shown in Figure 1, a power supply failure in one piece of equipment could propagate to another equipment by injecting out of specification voltages at the LVDS buffer terminals [3]. Due to the constraints of high speed signaling, it is not practical to use series protection resistors in the signal lines to reduce potential fault currents to an acceptable level; thus, it is necessary to add protection to the internal supply rails of each equipment.



Figure 1. Common mode voltages and fault propagation

The mitigation methods for both the common mode and fault propagation issues are time consuming to analyze for failure mode effects and they typically result in an increased complexity of the flight equipment.

It is thus highly desirable to incorporate galvanic isolation in the link paths; this will permit the legacy Mil-Std-1553B command and control links to be replaced with the more capable SpW bus and to eliminate failures in test environments with EGSE [4].

II. LIMITATIONS OF CURRENT ISOLATION SOLUTIONS

The Data and Strobe lines of SpaceWire are non-DCbalanced signal streams with data rates up to 400 Mb/s [1]. Since the signal streams are not DC-balanced, typical capacitive or inductive (transformer) AC-coupling methods for isolation are not viable (Fig. 2). In comparison, by design, high speed digital isolators are capable of handling non-DCbalanced signal streams; however, almost all high speed digital isolators available today have maximum data rates of less than 200 Mb/s. Although there are some LVDS digital isolators capable of data rates greater than 200 Mb/s, they are either not built on a space-proven, radiation-hardened process or, they cannot be implemented on readily-available, space-proven semiconductor platforms; Figure 3 illustrates the isolation of SpW receivers using LVDS digital isolators [4].



Figure 2. Standard Non-isolated SpW Link



Figure 3. SpW Receivers with LVDS Digital Isolation

III. DC-BALANCED SPACEWIRE

One approach to enabling the use of AC-coupling methods of isolation in SpW would be to add DC-balance encoding & decoding to a standard SpaceWire link [5][6]. However, the added complexity and overhead may affect the maximum data throughput rate or the maximum cable lengths or may even require an increase in the capabilities of the physical layer components. A typical implementation of DC-balanced SpW would be to add a DC-balance encoding stage on the transmitter side of a SpW link; this would allow the use of an AC-couple isolation stage at the receiver end of the SpW link followed by a DC-balance decoding stage prior to the standard SpW receiver (see Fig. 4).



Figure 4. DC-balanced SpW Link with AC-couple Isolation

Although this approach does enable isolation via ACcoupling, the data traffic across the SpW cable will be nonstandard; as such, it is not backward compatible with existing SpW networks. Depending on the intended method of achieving DC-balance, the encoder & decoder stage logic could be quite complex to implement [6].

IV. ICE-O TECH BIT-BALANCE FOR SPW ISOLATION

ICE-O Tech's innovative design approach encapsulates the DC-balance encoding & decoding stages along with the ACcoupling components into a functional isolation module/block at the receiver end of the SpW link (Fig. 5). While most DCbalance encoding schemes are implemented in the encoding layer [1][6], the Bit-Balance technique, developed by ICE-O Tech, is designed to achieve DC-balancing at the bit level in the physical layer; this distinction enables an efficient colocation of both the encoding & the decoding stages on either side of the AC-coupling stage. In this SpaceWire application, Manchester encoding was selected to extremely simplify the interim DC-balancing of Data; each bit is straightforwardly encoded to itself & its compliment ("0" => "01" and "1" => "10") to ensure an even balance of 0's & 1's. Although this encoding scheme represents a 100% overhead, it only exists within the confines of the functional isolation block; all points outside the isolation block remain standard SpW. The bitbalanced Data & the extracted Clock (inherently DC-balanced) are sent across the AC-coupled (inductive or capacitive) isolation stage and subsequently simply clocked back to the original standard SpW Data & Strobe signal streams (Fig. 6).



Figure 6. ICE-O Tech Isolation Block

V. BENEFITS OF THE ICE-O TECH ISOLATION APPROACH

The ICE-O Tech Isolation Block can be implemented using readily available space-proven components & platforms to achieve galvanically isolated SpaceWire with wide commonmode voltage tolerance; furthermore, the simple bit-level DCbalance encoding/decoding scheme can be efficiently configured as a small, low-power discrete solution or even potentially integrated into existing SpW interface components. As a potential drop-in SpW isolation solution (Fig. 5), no changes are needed to the standard SpW protocol, network topology, or bandwidth; this ensures backward compatibility with existing SpW networks and simplifies the planning & design of networks that may have both isolated & non-isolated SpW ports.

VI. DEMONSTRATION MODULE

To demonstrate the isolation capabilities of ICE-O Tech's innovative design approach, a demonstration (demo) module of an isolated SpW receiver will be built around the ICE-O Tech Isolation Block (Fig. 6). Two programmable logic devices will be used to implement the simple DC-balance encoding &

decoding stages separated by high speed isolating transformers or capacitors; an isolated DC-DC converter is also included to power the isolated side of the module (Fig. 7).



A pair of demo modules would be used to isolate the receivers at both ends of a SpW cable for testing.

VII. CONCLUSIONS

Adding DC-balancing to SpW enables the use of traditional capacitive or inductive isolation methods; encapsulation of the capacitor or transformer with both the simple encoding & straightforward decoding stages allows the localization of the DC-balanced non-standard SpW signaling. This approach provides a SpW isolation solution similar in topology to LVDS digital isolators [4] while utilizing commercial off-the-shelf (COTS) components and space-proven logic platforms.

Testing in SpW environments within the SpaceWire community is needed to confirm the effectiveness of the ICE-O Tech Bit-Balance isolation approach. The results from demo module testing will contribute to the definition and potential development of an optimized discrete SpW isolator module, an internal discrete on-board implementation, or a single multichip-module (MCM) implementation.

References

- European Space Agency ECSS Secretariat, "ECSS-E-ST-50-12C, Space engineering, SpaceWire – Links, nodes, routers and networks," 31st of July 2008, 129 pages
- [2] M. Suess, J. Ilstad, W. Gasti, "Galvanic Isolated SpaceWire Links, Requirements, Design Options and Limitations," 2009 ESA Workshop on Reliable Power & Signal Interfaces
- [3] R. Malmberg, "Failure Propagations via Power and Signal Interfaces," 2009 ESA Workshop on Reliable Power & Signal Interfaces
- [4] G. Baterina, Y. Moghe, H. Nimmett, and A. Senior, "Galvanic Isolation of SpaceWire Receivers," International SpaceWire Conference 2016
- [5] M. Epperly and S. Torno, "Galvanically Isolated SpaceWire," International SpaceWire Conference 2014
- [6] A. Kisin and G. Rakow, "New Approaches for DC Balanced SpaceWire," International SpaceWire Conference 2016

Radiation Tolerant Optical Transceivers for SpaceFibre Data Links

Components, Short Paper

Mikko Karppinen, Antti Tanskanen, Jyrki Ollila VTT Technical Research Centre of Finland Oulu, Finland mikko.karppinen@vtt.fi, antti.tanskanen@vtt.fi, jyrki.ollila@vtt.fi

Demetrio Lopez Molina, Juan Barbero Alter Technology TÜV NORD Tres Cantos - Madrid, Spain demetrio.lopez@altertechnology.com, juan.barbero@altertechnology.com

Cesar Boatella Polo, Richard Jansen, and Iain McKenzie European Space and Technology Centre / ESA Noordwijk, The Netherlands cesar.boatella.polo@esa.int, richard.jansen@esa.int, iain.mckenzie@esa.int

Abstract—Fiber-optic data links enable high-speed communications onboard satellites and spacecrafts. VTT has developed ruggedized optical transceivers for multi-gigabit interconnections in harsh environments, including also multi-lane links. The transceiver developed for the SpaceFibre optical physical layer supports full-duplex communication up to 6.25 Gb/s so far and is upgraded to 10 Gb/s. The hermetic packaging approach results in a compact and robust transceiver. It has passed stringent mechanical and thermal tests, as well as gamma radiation tests. Recently also heavy-ion and proton single-event-effect radiation tests were performed demonstrating that the transceiver is latch-up free and not sensitive to single event functional interrupts or destructive effects. Moreover, the effect of single bit-errors or error strings induced by energetic particles is estimated negligible in the in-orbit applications and with the normal encoding and error-correction schemes.

Index Terms— SpaceFibre, Optical Transceiver, Fiber-Optics, VCSEL, Radiation Single-Event Effect, Networking, Spacecraft Electronics.

I. INTRODUCTION

The data transmission capacity requirements onboard satellites and spacecrafts are rapidly increasing. Therefore, many new telecom, earth observation and other highperformance satellite payloads are expected to use fiber-optic interconnections. Fiber-optic data links enable clear advantages over the electrical cables. These include lightweight cabling insensitive to electromagnetic interference, galvanic isolation between the transmitter and receivers ends, and almost distance-independent communication performance. Therefore, fiber optics offers a generic interconnect solution that can cover both the backplane and inter-equipment communications.

Recently the SpaceFibre standard has been prepared for multi-gigabit, highly reliable data communications on-board satellite payloads [1]. SpaceFibre complements the capabilities of the widely used SpaceWire networking standard; improving the data rate by orders of magnitude, while massively reducing the cable mass. The Physical layer of the standard runs over electrical (i.e. copper) or fiber-optic cables. The fiber-optic physical layer is planned to support operation at bit-rates between 1...10 Gb/s per lane. The data links are proposed to operate at 850 nm optical wavelength and to use high-bandwidth multimode silica fibers as the transmission media.

The optical data links call for compact low-power optical transceivers ruggedized for the harsh space environment and the launching conditions. Although fiber optics has long been the main technology in terrestrial communications and in many other high-speed interconnects, it has emerged relatively slowly into space applications [2]–[5] and is not widely used there yet. The optical transceivers used in terrestrial applications do not meet all the requirements typical in satellite missions, such as, the temperature range, mechanical stability, hermeticity, and radiation tolerance.

VTT together with support from the European Space Agency and its project partners has developed a number of high-bit-rate optical transceivers to enable the optical physical layer, thus addressing the needs of the future payloads [5]–[8].

In this paper, we describe the properties, the functional performance and the environmental reliability of VTT's optical transceiver for SpaceFibre data links. Especially, we focus on the results of the latest heavy-ion and proton radiation tests that studied the single-event radiation effects on the data links.

II. OPTICAL TRANSCEIVER FOR SPACEFIBRE

A. Technology selections

Optical physical layer technology suggested for the SpaceFibre comprises of the transceivers based on the VCSEL (vertical-cavity surface-emitting laser) diodes transmitting at 850 nm wavelength a well as on the 50 μ m-core multimode graded-index (GI-MM) fiber cables. This is matured and well-established technology in many terrestrial very-short-reach interconnection applications today, as it supports multi-gigabit links with low power dissipation and relatively low costs.

Such 50-µm GI-MM fibers are available as radiationtolerant versions too [9] [10]. The bandwidth depends on the more detailed fiber design. Laser-optimized fibers can support data links reaching several hundreds of meters at 10 Gb/s. However, in the space environment, the maximum link distance (i.e. fiber length) is probably limited by the radiation induced optical loss of the fiber; thus, it depends on the mission specifications and on the transceiver and fiber properties.

B. Transceiver design

VTT has developed a VCSEL-based single-lane optical transceiver for space and other harsh environment applications including the SpaceFibre physical layer. The third generation transceiver, VTT's SolidOpto SPFI-003-6G, enables serial full-duplex communication at up to 6.25 Gb/s bit-rate and slightly beyond. This bit-rate matches with the latest generation high-speed serial interfaces of ASICs for space. The optical transceiver is protocol independent and includes no embedded data encoding. It has current mode logic (CML) type high-speed electrical inputs and outputs, and it needs a single +3.3 V power supply with a typical power dissipation of 210 mW.

The transceiver module is equipped with two $50/125 \,\mu\text{m}$ GI-MM fiber pigtail cables for the communication. Essentially the transceiver incorporates four active devices: a VCSEL laser with its driver IC in the transmitter side and a PIN photodiode (PD) with the receiver IC in the receiver side. The directly modulated GaAs VCSEL laser emits at 850 nm wavelength. The receiver IC includes trans-impedance and limiting amplifier (TIA and LIA) circuits. A schematical structure of the transceiver and the full-duplex optical physical layer link is illustrated in Fig. 1. The complete link consists of two transceiver modules and the fiber-optic cables connecting them. The fiber pigtail cables of the transceivers can be connected back-to-back, but often a longer reach link is made by applying patch cables between the pigtails.



Fig. 1. Schematics of the full-duplex optical data link using two transceivers.

The transceivers are built using fully hermetic metalceramic packaging technology. This results in compact, lightweight components combining high reliability, effective thermal management (without active control), as well as high electrical performance i.e. good signal integrity. The bare die photonics and IC devices are assembled on the multilayer ceramics substrates, more precisely on low-temperature cofired ceramics (LTCC) substrates. The receiver and transmitter optical subassemblies (ROSA and TOSA) are mounted on the main ceramics substrate that is also used as the module bottom. The GaAs PD and the receiver IC are located on the ROSA substrate, and the VCSEL is on the TOSA, whereas the driver IC is mounted on the main packaging substrate. For hermetic sealing of the whole assembly, a kovar frame and lid are attached on the main substrate. The hermetic optical fiber feedthroughs are sealed to the kovar frame by glass and metal soldering techniques. A photo of the transceiver is shown in Fig. 2. and a photo of the de-lidded transceiver in Fig. 3.

The transceiver module dimensions are $17 \times 17 \times 5 \text{ mm}^3$ and it has 48 contact pads on the bottom side. The module can be connected to a printed circuit board (PCB) by the use of a solderless electrical interposer.



Fig. 2. SolidOpto SPFI transceiver module (the fiber pigtails equipped with Diamond Mini-AVIM connectors).



Fig. 3. SolidOpto SPFI transceiver before sealing with metal lid (the fiber pigtails equipped with Radiall LuxCis connectors).

C. Functional Performance

The typical fiber-coupled optical output power of the transceiver is +2...2.5 dBm at room temperature, and the typical receiver sensitivity is -13 dBm at 6.25 Gb/s and at 10^{-12} bit-error rate (BER). This results in a large link margin supporting high reliability, as the margin is much larger than the initial optical loss of the fiber cables and connectors.

The optimized thermal management of the SpaceFibre transceiver enables good signal integrity even at high ambient temperatures (>85 °C), which are typically challenging for the VCSEL transceivers, mainly because of the inherent property of their VCSEL causing their optical output power to decrease rapidly at high temperatures.

Actually, the transmitter part is applicable up to 10 Gb/s and the active devices used in the receiver part would support

10 Gb/s operation too, but currently its applicability in space is limited to data rates below 7 Gb/s due to some electronics integration induced jitter. A new version of the transceiver under development includes an electronics design change that is expected to upgrade the receiver to be feasible up to 10 Gb/s.

III. ENVIRONMENTAL RELIABILITY

Stability and reliability of the SolidOpto SPFI transceivers have been investigated by environmental test campaigns. This chapter presents an overview of the most relevant earlier test results, whereas the more recent single-event radiation test results are presented in the following two chapters.

In addition to its robust mechanical structure, the transceiver exhibits several decibels of optical power budget margin (see previous chapter) against all kinds of environmental effects on the transceivers and the cables as well as against the materials degradation over the mission lifetime. The actual link margin depends on the specified operation temperature range, the data rate, and the fiber harness.

A. Thermal and Vacuum Testing

The typical storage temperature requirement of -55...+125 °C with long lifetime of the satellite mission causes a significant stress on the hermetic photonic package. SolidOpto SPFI transceivers with hermetic optical fiber feedthroughs have passed the thermal tests where a number of samples were stressed with temperature cycles between -55...+125 °C up to 1000 cycles and beyond. After the thermal stressing, no failure of the package or the optical feedthroughs were observed in the post-inspection, and the hermeticity of the packages was confirmed by gross and fine leak hermeticity tests. Standardized (MIL-STD 883 Method 1014.14) helium fine leak tests with rejection limit of 10^{-7} atm×cm³/s were applied.

In addition, thermal vacuum tests have been performed over the operation temperature range of -40...+85 °C without transceiver failures. The data link was running during the tests.

B. Mechanical Testing

The SpaceFibre transceivers have passed stringent mechanical vibrations and shock tests without failures. The mechanical shock tests were made to determine the effect of a half-sine wave shock of up to 3000 g acceleration at 10 kHz. The random vibration test were carried out up to 50 grms in 3 axis (according to MIL-STD-883 Method 2026). Complete electro-optical characterizations were carried out on each sample before and after the mechanical tests.

C. Gamma Radiation Testing

In satellite environment, the components are exposed to gamma radiation, which can result in a significant total accumulated dose during the lifetime. The transceivers were stressed in total ionizing dose (TID) radiation tests with gamma radiation up to 100 kRad total dose level at the dose rate of 541 rad/h (tested in the CIEMAT facility). The transceivers were powered with the normal 3.3 V supply during the exposure. No notable degradation of the optical power or the transceiver performance could be observed during the TID tests.

IV. HEAVY-ION RADIATION TESTING

Since the transceiver incorporates two ICs and the semiconductor photonics devices, the radiation induced single event effects (SEE) may influence the reliability and the biterror rate performance of the data link in the space environment. Therefore, both heavy-ion and proton radiation tests have been performed looking specifically at SEE on both the transmitter and receiver sections of the transceiver.

A. Heavy-Ion Test Setup and Conditions

The SolidOpto SPFI-003-6G optical transceiver has been radiation tested at the heavy-ion synchrotron CYCLONE in Louvain la Neuve, Belgium to characterise its SEE performance. The test setup is shown in Fig. 4. Two de-lidded optical transceivers on their evaluation boards are present in the radiation chamber, as shown in Fig. 5. They are supplied from a latching current limiter power supply to protect the devices against potential latch-up events. A 5 Gb/s signal is generated from a Xilinx FPGA evaluation board ML605 that is configured as a BER tester. Its signal drives the VCSEL driver of the first transceiver that is linked with an optical fiber to the receiver of the second optical transceiver, where the TIA-LIA receiver IC feeds the signal back to the BER tester. The VCSEL driver is tested by exposing the first optical transceiver and the photodiode and TIA-LIA receiver IC are tested by exposing the second device. For the receiver, the beam was tilted to allow the ions to pass the kovar package rim and reach the IC and the PD. During the radiation test, the bit errors are recorded and subsequently divided by the fluence to determine the bit error cross-section. The radiation beam test conditions are presented in Tables I and II.



Fig. 4. Heavy-ion radiation test set-up



Fig. 5. Photo of the heavy-ion radiation test set-up including two optical transceiver samples on their evaluation boards

Ions	Angle (deg)	LET (MeVcm ² /mg)	Fluence (cm ⁻²)
Ne-22	0	3.3	12 x10 ⁶
Ar-40	0	10	1 x10 ⁶
Ni-58	0	20.4	1 x10 ⁶
Kr-84	0	32	10 x10 ⁶
Xe-129	0	62.5	10 x10 ⁶

 $TABLE \ I. \ \text{Heavy-ion BER Radiation Test Conditions for the} \\ \text{Receiver Including both the Photodiode and the Receiver IC} \\$

 TABLE II.
 Heavy-ion BER Radiation Test Conditions for the VCSEL Driver

Ions	Angle (deg)	LET (MeVcm ² /mg)	Fluence (cm ⁻²)
C-13	45	1.8	20 x10 ⁶
Ar-40	45	14.1	1 x10 ⁶
Ni-58	45	28.9	1 x10 ⁶
Kr-84	35	39.1	1 x10 ⁶
Kr-84	45	45.3	10 x10 ⁶
Xe-129	45	88.4	10 x10 ⁶

B. Heavy-Ion Test Results

For the latch-up radiation tests the SolidOpto SPFI-003-6G optical transceiver was exposed to Xe ions at an angle of 45 degrees, which corresponds to an equivalent linear energy transfer (LET) of $88.4 \text{ MeVcm}^2/\text{mg}$ and a fluence of $20 \times 10^6 \text{ cm}^{-2}$ at a temperature of $28 \text{ }^\circ\text{C}$. No latch-up was detected, thus the device can be considered to be latch-up free.

For the single-event transient (SET) radiation tests to determine the bit error cross-section as a function of radiation flux, the ions used and respective LET in the VCSEL driver and receiver are shown in Tables I and II. The measured bit error cross-section at 5 Gb/s as a function of LET for the receiver and VCSEL driver is shown in Fig. 6.



Fig. 6. The bit error cross-section for the receiver (top) and transmitter (bottom) of the SolidOpto SPFI-(003)-6G optical transceiver

The LET threshold for the receiver part, including the photodiode and the TIA-LIA receiver IC, has been found to be $0.7 \text{ MeVcm}^2/\text{mg}$ with a bit error saturation cross-section of $4.8 \times 10^{-4} \text{ cm}^2$ bit error at 5 Gb/s. For the VCSEL driver, the LET threshold is found to be 1 MeVcm²/mg and the bit error saturation cross-section found is $1.7 \times 10^{-4} \text{ cm}^2$ bit error at the same data rate. The longest SET pulse detected for the receiver was 85 ns and for the VCSEL driver 14 ns. The low LET threshold required also proton radiation tests to be carried out to determine the proton radiation cross-section accurately.

V. PROTON RADIATION TESTING

Potential single-event effects on the transceiver were studied with proton radiation exposure too.

A. Proton Test Setup

The proton tests were performed at the Proton Irradiation Facility at the Paul Scherer Institute in Villingen. The radiation test set-up is shown in Fig. 7. The long distance between the radiation chamber and the control room required the output of the BER tester to be transmitted via optical fiber to the optical transceiver in the radiation chamber to ensure that the tests could be performed at 5 Gb/s data rate. The signal of the Centellax TG1B1-A 10G BER tester was transmitted via the SolidOpto SPFI-003-6G optical transceiver to the device under test in the radiation chamber. There the optical signal is received by the TIA-LIA receiver IC and electrically relayed to the second optical transceiver where the signal is transmitted via the VCSEL driver back to the BER tester. The transceivers in the chambers were protected from potential latch-up events by a latching current limiter power supply.



Fig. 7. Proton radiation test-setup

B. Proton Test Results

For the VCSEL driver radiation test the maximum proton energy 200 MeV was used for a combined fluence of $20x10^{10}$ cm². The two VCSEL drivers tested did not show any SET sensitivity for the highest proton energy and can therefore be considered proton radiation insensitive.

For the receiver BER proton radiation tests, four different proton energies have been selected; 50, 100, 150 and 200 MeV for a fluence of 1×10^{10} cm⁻². For proton beam incidence angles

up to 85 degree from the normal a worst case bit error crosssection of 7.7×10^{-9} cm² has been measured. Detailed investigation of the SETs indicates that in excess of 95% of bit errors arise from the protons impinging on the photodiode and are less than 500 ps of duration. The remaining SETs arise from the TIA-LIA receiver IC and are less than 14 ns long.

VI. TOWARDS HIGHER BIT-RATE MULTI-LANE TRANSCEIVERS

VTT has also developed parallel optic transceivers, for instance, for high-throughput onboard processors of telecom payloads and for multi-laning of SpaceFibre links, thus enabling higher aggregate data rate and integration density. Recently, novel transceivers were developed based on 6-core multimode fibers and co-designed 6-channel optoelectronic engines operating at 25 Gb/s channel rates [8] [11]. In the collaborative project, the technologies were optimized for low power, wide temperature range, and radiation tolerance. A photo of the demonstrated mid-board multi-lane transceiver for space is presented in Fig. 8.



Fig. 8. Multi-lane transceiver employing 6-core fibers on evaluation board

VII. CONCLUSIONS

The presented 6.25 Gb/s optical transceiver provides the performance and functionality needed for the use in the SpaceFibre data links. It meets the reliability and stability requirements for the harsh satellite environment applications. The heavy-ion and proton radiation tests showed that the transceiver is latch-up free, and no functional interrupts (SEFI) nor destructive radiation events has been detected.

Nevertheless, heavy ions can cause single-event transients (SET) that may induce moderately long bit-error strings in both receiver and transmitter. In addition, protons impinging the photodiode may cause bit-errors. Anyway, these single events are rare in the satellite applications and even with no encoding or error correction coding the BER cross-section is very small. It is predicted that with encoding and limited error correction coding the radiation induced BER drops to a negligible level, i.e. below the inherent BER of the data link.

A slightly modified version of the transceiver is currently implemented increasing the bandwidth to 10 Gb/s operation. Since it is using exactly the same active semiconductor devices and materials, it will have the same radiation tolerance. VTT is also developing radiation tolerant multi-lane transceiver modules up to 25 Gb/s channel rates.

ACKNOWLEDGMENT

The authors wish to thank all other project partners on the SpaceFibre optical data link studies, especially the colleagues at Fiberpulse, Patria, Radiall, and Thales Alenia Space.

VTT's activities towards the higher bit-rate multi-lane transceivers has received funding from the European Union FP7 program under the grant number 607274 (ROBIN project). The authors from VTT thank all ROBIN project partners.

REFERENCES

- [1] S. Parkes, C. McClements, D. McLaren, A. F. Florit, and A. G. Villafranca, "SpaceFibre: A multi-Gigabit/s interconnect for spacecraft onboard data handling," IEEE Aerospace Conference, 7–14 March 2015, Big Sky, MT, USA, pp. 1–13.
- [2] M. N. Ott et al., "Applications of optical fiber assemblies in harsh environments: The journey past, present, and future," Proc. SPIE Vol. 7070 (2008).
- [3] K. Kudielka, P. Boehle, M. Tornell, and A. Borges Alejo, "Design, development, and verification of the fibre-optic harness for SMOS," Elektrotechnik & Informationstechnik 124(6), 175–179, (2007).
- [4] Y. Chen, H. Hemmati, and R. Some, "Multi-Gb/s Fiberoptic Physical Layer for Spacecraft Interconnects," Journal of Lightwave Technology, Vol. 31 (12), pp. 1899–905 (2013).
- [5] N. Karafolas, J. Armengol, and I. Mckenzie, "Introducing photonics in spacecraft engineering: ESA's strategic approach", Proc. of IEEE Aerospace Conf., 7–14 March 2009, Big Sky, MT, USA.
- [6] M. Karppinen, V. Heikkinen, K. Kautio, J. Ollila, and A. Tanskanen, "Fiber-Optic Transceivers for High-Speed Digital Interconnects in Satellites," Proc. of Int. Conf. on Space Optics (ICSO), 7–10 Oct 2014, Tenerife, Spain.
- [7] N. Venet et al., "High-throughput optical inter-board interconnects for next generation on-board digital transparent processors," Proc. of Int. Conf. on Space Optics (ICSO), 7–10 Oct 2014, Tenerife, Spain.
- [8] M. Karppinen et al., "Integration of 150 Gbps/fiber optical engines based on multicore fibers and 6-channel VCSELs and PDs," SPIE Optical Interconnects XVI, Feb 2016, San Francisco, CA, USA, Proc. SPIE Vol. 9753 (2016).
- [9] M. Alam, J. Abramczyk, J. Farroni, U. Manyam, and D. Guertin, "Passive and Active Optical Fibers for Space and Terrestrial Applications," SPIE Photonics for Space Environments XI, Proc. of SPIE Vol. 6308 (2006).
- [10] E. Regnier et al., "Recent Developments in Optical Fibers and How Defense, Security and Sensing can Benefit," ISROS 2009, 16 May 2009, Cagliari, Sardinia (Italy).
- [11] M. Ko, K. Tittelbach-Helmrich, V. Petrovic, and D. Kissinger, "25-Gb/s/channel VCSEL driver and transimpedance amplifier array ICs in 0.25-µm SiGe:C BiCMOS technology for space applications", Proc. of AMICSA 2016, 12–16 June 2016, Gothenburg, Sweden.

Programmable SpaceFibre interface with NanoBridge Field Programmable Gate Array

Components, Short Paper

Hiroki Hihara^{1,2}, Mitsunobu Kuribayashi¹, Kyoko Murozono¹, Kazuyuki Yamada¹ ¹Space Engineering Division NEC Space Technologies, Ltd. Tokyo, Japan {h-hihara@bc, m-kuribayashi@cp, k-murozono@ak, k-yamada@ea}.jp.nec.com Yu Otake², ²Space Systems Division NEC Corporation Tokyo, Japan y-otake@bp.jp.nec.com

Kazutoshi Kobayashi³, Yuto Tsukita³, Haruki Maruoka³, Jun Furuta³ ³Kyoto Institute of Technology Kyoto, Japan kazutoshi.kobayashi@kit.ac.jp

Abstract-SpaceFibre draft standard provides a common interface specification among various Serializer/Deserializer (SERDES) interface devices for high data rate transmission. Since programmability is indispensable to exploit the versatility of SpaceFibre standard, we developed a rewritable field programmable gate array (FPGA) with radiation hardened characteristics using atom switch, which is called NanoBridge. NanoBridge provides rewritable capability for FPGAs without static random access memory (SRAM) or electrically erasable programmable read-only memory (EEPROM) to store configurations, which results in mitigating soft-errors in configuration memories. Development tools using high level synthesis technology was also developed to enhance programmability. Memory patrol and memory scrubbing functions are not required for a NanoBridge FPGA, and that eliminates external peripheral devices used with conventional rewritable FPGAs. Irradiation test and thermal cycling test of NanoBridge has been completed successfully. The demonstration of the device in orbit is planned in 2018.

Index Terms— SpaceFibre, Field Programmable Gate Array (FPGA), Atom Switch, Non Volatile, Rewritable, High Level Synthesis, Behavioral Synthesis, Dynamically Reconfigurable Architecture.

I. INTRODUCTION

High resolution and multi-spectral optical sensors, and high resolution synthetic aperture radars (SARs) are most demanding mission payloads nowadays. These payloads produce large volume of data that require high speed transmission capacity through onboard network within satellites. High speed data transmission interfaces over a few gigabits per seconds have already been provided, whereas no common interface specification among devices manufactured by various vendors had not been provided. SpaceFibre draft standard [1] is the first practical interface specification to provide a common interface specification among various Serializer/Deserializer (SERDES) interface devices for gigabit data transmission. Since SpaceFibre standard provides routing and virtual channel transmission functions, programmability is indispensable to exploit the versatility.

FPGAs are often used for the onboard electronics for space crafts as satellites [2]. High reliability is necessary to the circuitries of FPGAs used in space. A soft error is a temporary breakdown caused by cosmic radiation, and the main cause of soft errors in space is heavy ions. When radiation sources collide with an integrated circuit, stored values in storage cells are often flipped, and the flipped data could cause malfunctions. A countermeasure against soft errors is inevitable because data stored in flip-flops (FFs) of FPGA and/or static random access memories (SRAMs) are often flipped due to cosmic radiation. Thus radiation hardness is essential, because the amount of data loss caused by a single event effect (SEE) of cosmic radiation is not negligible on high speed and large volume data transmission using SERDES interfaces. We developed a rewritable field-programmable gate array (FPGA) with radiation hardened characteristics using atom switches, which is called NanoBridge [3]. Atom switches are independent of specific FPGA architectures, and thus it can take over electrically erasable programmable read-only memories (EEPROMs) or SRAMs in various conventional FPGAs to provide configuration information. Atom switches provide rewritable capability for FPGAs without EEPROM nor SRAM to store configuration data, which results in mitigating softerrors in configuration memories of conventional rewritable FPGAs. Memory patrol and memory scrubbing functions are

not necessary for a NanoBridge FPGA, and that eliminates external peripheral devices used with conventional rewritable FPGAs. In consequence, low power consumption can be achieved, and it is also the merit of NanoBridge when it is used for SERDES interfaces, because high speed interface devices consume much power. NanoBridge can add programmable capability to the interfaces of high resolution/multi-spectral optical sensors and SARs for which a dedicated applicationspecific integrated circuit (ASIC) was developed to mitigate soft-errors and to reduce power consumption.

Irradiation test and thermal cycling test of NanoBridge has been completed. Radiation hardened CMOS primitive cells are under development to provide NanoBridge as libraries. The demonstration of the FPGA in orbit is planned in 2018. We also replaced FFs with soft-error resilient ones to enable rewriting and to realize highly reliable FPGAs in space [4]. The evaluation results of the soft error resilience against heavy ions of NanoBridge and BCDMR-FF (Bistable Cross-coupled Dual Modular Redundancy FF) [4] which is a soft error tolerant FF are described in this paper.

II. RELATED WORK AND PROBLEM FORMULATION

We implemented and evaluated the SpaceFibre [5] to realize high speed data transmission required for high resolution and multi-spectral optical sensors, and high resolution synthetic aperture radars (SARs). As a result, we identified that wire-rate processing capability is mandatory for the sensor interfaces. Wire-rate processing speed is required on high speed signal outputs of onboard high resolution sensors, although applying programming capabilities are anticipated. It is not practical to use a high speed Micro-Controller Unit (MCU) for each sensor signal output due to its high power consumption and its large footprint. Dynamically reconfigurable processors (DRPs) using high-level synthesis were proposed [6, 7, 8] to improve the efficiency, whereas the inefficiency of fixed mesh is pointed out [9]. Fixed bit widths of data paths, elementary blocks and switch matrices aiming at mass production of the devices was one example of the inefficiency. Sensors have various data interfaces, such as 8, 12, 14, or 16 bits. Predefined data path between arrays of arithmetic logic units (ALUs) prevents behavioral synthesis tools from the optimization of layout size and the reduction in power consumption. Therefore, optimized ALUs and flexible data paths are required to embed processors in sensor units.

A rewritable FPGA is an alternative of an MCU and a DRP to apply programming capabilities to a sensor interface, because an FPGA can realize wire-rate processing speed in addition to add programming capabilities. However, the inefficiency of placement and routing (P&R) using configurable logic blocks (CLBs) and look-up tables (LUTs) with switches, together with associated configuration memories, is pointed out [9]. Various optimization techniques for the processing of sensing images were summarized and proposed [10, 11, 12]. P&R as area-optimal mapping was also mentioned, whereas these improvements were based on the optimization of scheduling and did not exploit high-level synthesis technology.

Design re-use of macro blocks used in high-level synthesis was proposed to optimize P&R process [13]. The reported method is applied to the layout process after logic synthesis. Code generation optimization using high-level synthesis was reported [14], while it is used for the optimization of streaming pipelines. In addition to that, single event effects caused by cosmic radiation are not negligible when SRAMs and EEPROMs are used as configuration memories.

We identified two issues missed in conventional rewritable FPGAs. The first requirement is to mitigate SEEs in configuration memories caused by cosmic radiations. The second issue is the adaptation of high-level and behavioral synthesis technology to accommodate programmability without preventing wire rate signal processing capabilities through high speed interfaces. Bit width should be optimized for specific sensors while maintaining the efficiency of calculations

III. DEVELOPMENT STATUS AND TEST RESULTS

NanoBridge FPGA [3] was proposed to realize signal processors for onboard sensors. NanoBridge can provide programming capabilities without using SRAMs nor EEPROMs for configuration information. In consequence, radiation hardness against SEEs is realized. Complementary metal oxide semiconductor (CMOS) circuits should be radiation hardened, in spite of the SEE free characteristics of NanoBridge, because CMOS circuits are the same as ordinary ASICs. We proposed a new type of flip-flop, which is called BCDMR-FF [4] to strengthen the radiation hardness of CMOS layers. Radiation tests were performed for the evaluation of NanoBridge and BCDMR-FF. As for NanoBridge, we developed test chips using a 40 nm CMOS technology. Test chips were also developed for the evaluation of BCDMR-FF using a CMOS 65 nm technology.

As for NanoBridge, the fabricated atom-switch FPGA includes a 40 x 40 logic cell array, each of which consists of a switch block and a logic block including 4 x 4-input LUTs. The 4.38-Mb atom switches are integrated between 4th and 5th Cu metal wires interconnect using 40 nm CMOS technology. To achieve a high OFF state reliability, two atom switches are serially connected with opposite direction in each element. Two atom switches are programmed by using the cell transistor connected to the middle node [15]. This switch is named as a complementary atom switch (CAS), since the voltage stress is shared by two atom switches complementarily. Small area and small capacitance (~0.14fF) of CAS contributes to low power consumption and high speed of FPGA. Its non-volatility also allows an immediate wake-up/sleep operation without reloading the configuration data, and saves the standby power effectively.

We compared the performance of the atom-switch FPGA with the commercial SRAM-based FPGA, which was a low-power one (iCE40) for mobile applications by Lattice Corp. [16]. The benchmark circuit of 16-bit Arithmetic Logic Unit (ALU) was mapped on both FPGAs. The atom-switch FPGA operated at 3.8 times faster clock frequency over the conventional one. The dynamic and active power consumptions of the atom switch FPGA were reduced by 67%

and 39%, respectively. These improvements are mainly originated from small capacitance of atom switch and smaller logic-cell size, which is a half of the conventional one at same technology node of 40nm.

Radiation resistance of atom switch was evaluated by using a heavy ion cocktail beam. For this evaluation, array of 128kbit atom switches or atom-switch FPGA were used. The array of atom switches were exposed by Xe and Kr ions. Linear energy transfers (LETs) of Xe and Kr were estimated to be 68.9 and 40.3 MeV/(mg/cm²) at the chip surface, respectively. During the irradiation, we observed no SEE, showing that SEE cross section is at least 100 times lower than that of NAND flash [17].

We made test element group (TEG) chips of D-FFs with reset terminals and BCDMR-FFs with reset terminals by using a 65 nm FDSOI (Fully Depleted Silicon on Insulator) process. The measurement evaluation of soft error resilience caused by heavy ions in space was performed at Takasaki Advanced Radiation Research Institute. Measurement conditions are shown Table I. Duplicated latches and C-elements are implemented in BCDMR-FF. Even if one data within those elements is reversed, an original value is maintained. Circuit diagrams of D-FF and BCDMR-FF are shown in Fig. 1 and 2, respectively. The areas of the FFs are shown in Table II.

TABLE I. MEASUREMENT CONDITIONS

Power supply voltage	0.8 V
Beam irradiation time	30 s
Ion beam source	Ar : 17.5 MeV, Kr : 40.9 MeV
Measurement times	20 times / each condition
Flux	Ar : 1.65×109 particle/cm ² Kr : 5.74×107 particle/cm ²

TABLE II. THE AREA OF D-FF AND BCDMR-FF

FF	Area	Area ratio
D-FF	$9.8\mu m^2$	1
BCDMR-FF	$24.5 \mu m^2$	2.25

Measurement result is shown in Fig. 3. No soft error was observed against Ar and Kr beams on BCDMR-FFs. The error bars were calculated with 68 % of reliability region because the number of errors was zero. More than 50 times of resilience against Ar and 80 times against Kr were observed for BCDMR-FFs compared with D-FFs. An area overhead of BCDMR-FF is 2.25 times as large as a conventional D-FF, whereas it has sufficient reliability for space system applications.

As for programmability, we propose a new design framework with four-layer architecture to design embedded processing elements (PEs) in the signal processing equipment of onboard sensors using a high-level and behavioral synthesis tool. It exploits the repetitive high-level synthesis process. Macro blocks synthesized through high-level and behavioral synthesis are registered in a database before the system level synthesis, and the information in the database is used for the optimization of resource consumption through the system level synthesis. We investigated the dependencies of resource consumption on the granularity of coarse grained function definitions using the extended database management function of ELEGANT tool chain [18, 19, 20]. The evaluation results show that small footprint was achieved especially with dynamically reconfigurable technique.



Fig. 1. Circuit diagram of D-FF with reset terminal



Fig. 2. Circuit diagram of BCDMR-FF with reset terminal

The purpose of the conventional high-level synthesis tools is to generate finite state machines with data paths (FSMDs). We refer to FSMD as fine grained layer. We expanded the scope of high-level synthesis to coarse grained layer to exploit FPGAs. Arithmetic-logic units (ALUs) in coarse grained layer are defined by high level programming languages. We also added switches for the scope of high-level synthesis. SpaceFibre routing switches can be assigned as switches highlevel synthesis. A four-layered structure is shown in Fig. 4 [20]. The layers consist of I/O circuitries, a fine grained layer, a coarse grained function definition layer and a switch layer, where they are listed from the bottom layer to the top layer, respectively. The following explains these layers.



Fig. 3. Cross section by heavy ion tests



Fig. 4. Circuit diagram of D-FF with Reset

I/O circuitries are implemented with random logic gates and mixed signal I/O circuitries. They are connected to the fine grained blocks in the above layer. The fine grained layer mainly consists of finite state machines and data paths. They are replaceable to follow the context described by high level languages. The coarse grained function definition layer is located on the fine grained layer. Optimized ALUs for a certain application are defined in this layer. An operation primitive defined in the layer corresponds to an operation code (opcode) of a conventional MCU, whereas it does not have to be standardized for over many applications. As for the topmost switch layer, SpaceFibre routing switches can be placed over coarse grained blocks. The connections between inputs and outputs of PEs are configurable with the switches.

The following six steps compose the design flow of the design framework to design a PE through layered scheme. [Step 1]: Describe a system in high-level language.

[Step 2]: Define coarse grained operations. Typical dedicated functions for sensor signals are signal compensation, feature point identification for image recognition, and image compression in addition to basic arithmetic operations. The defined coarse grained operations are exploited in high-level synthesis and behavioral synthesis process.

- [Step 3]: Generate source codes written in hardware description language (HDL) through behavioral synthesis. The hard macros defined and implemented in the step 2 are exploited for generating circuitries by the functions of CyberWorkBench (CWB) [21].
- [Step 4]: This step is identical to conventional logic synthesis.
- [Step 5]: This step is identical to conventional layout design.
- [Step 6]: The verification and validation step include back annotation based on the result of delay analysis.

We developed source code level debugger for C language programming environment. We developed Graphical User Interface (GUI) based source code debugger to debug the circuitries implemented in FPGAs in a high level language as C language. Evaluation boards are also provided to debug circuitries on the fly. GUI interface and an evaluation board is shown in Fig. 5.



Fig. 5. Source code debugging in C-language and FPGA evaluation board

We used a data base management function of CWB to exploit the definitions of operations in the coarse grained layer during the behavioral synthesis to treat a function as an operator. The operations can be implemented as hard macros. The operator definitions were exploited on the step 3 as macro blocks. They are registered in a database of CWB. Once macro blocks are registered in a database, CWB can use the macro blocks during the binding process of high-level synthesis automatically to reduce layout area size. The binding process of high-level synthesis is regarded as NP-hard and heuristic approach was often employed. The design flow enables the automation of binding process instead of the heuristic approach.

IV. CONCLUSION

SpaceFibre is the first practical common interface specification among various SERDES devices used for high resolution and multi-spectral optical sensors and SARs. We developed a programmable ASIC to exploit the versatility of SpaceFibre. It is realized with atom switches without configuration memories, and thus low error rate over gigabit transmission interfaces is expected. It will be evaluated in orbit via the innovative satellite technology demonstration program of JAXA in 2018. During a whole year, the full-HD image will be compressed and transmitted to ground stations.

ACKNOWLEDGMENT

The chip fabricated in a 65 nm is supported by VLSI Design and Education Center (VDEC), the University of Tokyo in collaboration with Renesas Electronics, Synopsys, Cadence Design Systems, and Mentor Graphics, Inc.

Authors thank T. Sugibayashi, M. Miyamura, T. Sakamoto, and M. Tada of NEC Central Research Laboratories for providing us Intellectual Property of Nanobridge. Authors also thank Dr. Makimoto for his precious advice regarding Highly Flexible Super Integration (HFSI) shown in the technology trend in Makimoto's wave [22].

REFERENCES

- ECSS-E-ST-50-11C-DFR1, "SpaceFibre Very high-speed serial link," ECSS Secretariat, ESA-ESTEC, Requirements & Standards Division, 2018.
- [2] C. Urbina-Ortega, G. Furano, G. Magistrati, K. Marinis, A. Menicucci, and D. Merodio-Codinachs, "Flash-based FPGAs in Space, design guidelines and trade-off for critical applications," 2013 14th European Conf. on Radiation and Its Effects on Components and Systems (RADECS), 2013, pp. 1–8.
- [3] M. Miyamura, T. Sakamoto, X. Bai, Y. Tsuji, A. Morioka, R. Nebashi, M. Tada, N. Banno, K. Okamoto, N. Iguchi, H. Hada, T. Sugibayashi, Y. Nagamatsu, S. Ookubo, T. Shirai, F. Sugai, and M. Inaba, "NanoBridge-Based FPGA in High-Temperature Environments," IEEE Micro, vol. 37, Issue 5, 2017, pp. 32 42.
- [4] J. Furuta, C. Hamanaka, K. Kobayashi, and H. Onodera, "A 65nm Bistable Cross-coupled Dual Modular Redundancy Flip-Flop capable of protecting soft errors on the C-element," in Proc. 2010 Symposium on VLSI Circuits, 2010, pp. 123 – 124.
- [5] Y. Otake, K. Hosokawa, Y. Sota, T. Tanaka, and H. Hihara, "Performance evaluations and proposal to improve nextgeneration SpaceFibre protocol," in Proc. of Intl. SpaceWire Conf. 2013, 2018, pp. 271 – 275.
- [6] M. Meribout and M. Motomura, "A Combined Approach to High-Level Synthesis for Dynamically Reconfigurable Systems," IEEE Trans. on Computers, vol. 53, no. 12, 2004, pp. 1508-1522.
- [7] T. Toi, N. Nakamura, Y. Kato, T. Awashima and K. Wakabayashi, "High-Level Synthesis Challenges and Solutions for a Dynamically Reconfigurable Processor," in 2006 IEEE/ACM Intl. Conf. Computer Aided Design, 2006, pp. 702-708.

- [8] Y. Hasegawa, S.Tsutsumi, V.Tanbunheng, T.Nakamura, T.Nisimura, and H.Amano, "Design Methodology and Tradeoffs Analysis for Parameterized Dynamically Reconfigurable Processor Arrays," in Proc. of FPL 2007, 2007, pp. 796-799.
- [9] R. Hartenstein, "The Microprocessor Is No Longer General Purpose: Why Future Reconfigurable Platforms Will win," in Proc. Second Annual IEEE Intl. Conf. on Innovative Systems in Silicon, 1997, pp. 2-12.
- [10] R. Zhao, M. Tan, S. Dai, and Z. Zhang, "Area-efficient pipelining for FPGA-targeted high-level synthesis," 2015 52nd ACM/EDAC/IEEE Design Automation Conference (DAC), 2015, pp. 1-6.
- [11] F. Wang, G. Wang, R. Wang, and X. Huang, "FPGA implementation of Laplacian of Gaussian edge detection algorithm," Advanced Materials Research, vol. 282-283, 2011, pp. 157-160.
- [12] L. Chen, "Fast Generation of Gaussian and Laplacian Image Pyramids Using an FPGA-based Custom Computing Platform," Master degree thesis, Virginia Polytechnic Institute and State Univ., 1994.
- [13] M. Gort, and J. Anderson, "Design re-use for compile time reduction in FPGA high-level synthesis flows," 2014 Intl Conf. on Field-Programmable Technology (FPT), 2014, pp. 4-11.
- [14] M. Schmid, O. Reiche, C. Schmitt, F, Hannig, and J. Teich, "Code Generation for High-Level Synthesis of Multiresolution Applications of FPGA," in Proc. of FSP2014, 2014, pp. 21-26.
- [15] M. Tada, N. T. Sakamoto, Banno, K. Okamoto, N. Iguchi, H. Hada, and M Miyamura, "Improved ON-State Reliability of Atom Switch Using Alloy Electrodes," IEEE Trans. on Electron Devices vol. 60, no. 10, 2013, pp. 3534-3540.
- [16] X. Bai, T. Sakamoto, M. Tada, M. Miyamura, Y. Tsuji, A. Morioka, R. Nebashi, N. Banno, K. Okamoto, N. Iguchi, H. Hada, and T. Sugibayashi, "A low-power Cu atom switch programmable logic fabricated in a 40nm-node CMOS technology," Symp. on VLSI Technology, Dig. Tech. Papers, 2017, pp. T28-T29.
- [17] K. Takeuchi, "Feasibility study for the ultra-low energy consumption, high Speed, non-volatile Technology", The 28th Microelectronics Workshop (MEWS28), Tsukuba, 2015.
- [18] K. Nakagawa, T. Kinoshita, O. Takeda, M. Tsuji, and T. Yamamoto, "Result of Preliminary design on Electric Design Guidance Tool for Space Use," JAXA Research and Development Memorandum, 2005.
- [19] H. Hihara, Y. Nishihara, M. Nomachi, T. Takahashi, and T. Takashima, "Designing Space Cube 2 with ELEGANT framework," in Proc. of Intl. SpaceWire Conf. 2008, 2008, pp. 219 222.
- [20] H. Hihara, A. Iwasaki, M. Hashimoto, H. Ochi, Y. Mitsuyama, H. Onodera, H. Kanbara, K. Wakabayashi, T. Sugibayashi, T. Takenaka, H. Hada, M. Tada, M. Miyamura, and T. Sakamoto, "Sensor signal processing using high-level synthesis with a layered architecture," IEEE Embedded Systems Letters, vol. PP, Issue 99, 2018, pp. 1 – 1.
- [21] K.Wakabayashi, "CyberWorkBench: Integrated design environment based on C-based behavior synthesis and verification," in Proc. IEEE VLSI-TSA Intl. Symposium, 2005, pp. 173-176.
- [22] T. Makimoto, "Implications of Makimoto's Wave," IEEE Computer, vol. 46, no. 12, 2013, pp. 32 – 37.

On-board Equipment & Software (Short)

SpaceWire and SpaceFibre for Internal Architecture in NVDRs

On-board equipment and software, Short Paper

Toru Sasaki

Mitsubishi Electric Corporation (MELCO), Advanced Technology R&D Center 8-1-1, Tsukaguchi-Honmachi, Amagasaki, Hyogo, Japan Sasaki.Toru@eb.MitsubishiElectric.co.jp

Abstract— In this study, we report a way to use SpaceWire in NVDRs (Nonvolatile Data Recorders) and SpaceFibre study for a next-generation NVDR. We already reported the proto-type development of an NVDR in International SpaceWire conference 2013. Since then, we have upgraded the design for some flight models. In the NVDR, SpaceWire is used only for an internal control network. Payload data internal links and the control network are separated. NVDR software controls external interfaces and memory controllers using our dedicated protocol like RMAP over SpaceWire. We could adopt some techniques for optimization because interoperability is not needed for the internal control network. First, we used LVTTL for SpaceWire instead of LVDS. Next, we implemented a 1-to-N router instead of a cross-bar switch router. Finally, to save more connector pins, we converted SpaceWire into SPI (Serial Peripheral Interface). During the hardware and software development, we obtained some lessons learned for SpaceWire implementation. In addition to the NVDR development, we are now designing a nextgeneration NVDR which has more high data rate capability and lager capacity at more compact size. In the next-generation NVDR we have a plan to use Microsemi's RTG4 because RTG4 has several High-Speed serial interfaces and SpaceWire dedicated interfaces. SpaceFibre is one option for internal payload data links. Moreover, there is a possibility that the control network can be synthesized to SpaceFibre as one of virtual channels. We are studying the internal architecture including both the control network and the internal payload data links. We report the baseline architecture and a way to configure a SpaceFibre multi-lane link.

Index Terms— SpaceWire, SpaceFibre, Data Recorder, Mass memory, Spacecraft Electronics.

I. INTRODUCTION

Space missions for earth observation have required high data rate and large capacity data recorders for spacecrafts [1]. Melco have developed high speed and large capacity data recorders (NVDRs) for these advanced requirements. NVDRs use NAND flash memories to satisfy the requirements. Because NAND flash memories require more complicated control than other types of memory, we introduced SpaceWire and a dedicated protocol for the control. This NVDR project is based on the development supported by JAXA from 2009 to

Shinya Hirakuri

Mitsubishi Electric Corporation (MELCO) Kamakura Works 325, Kamimachiya, Kamakura, Kanagawa, Japan Hirakuri.Shinya@da.MitsubishiElectric.co.jp

2013, and we have brushed up the architecture for higher specifications. We already reported the past development in SpaceWire conference 2013 [2]. Since then, we needed to upgrade NVDR specifications for greater demands of earth observation satellites. During the refinement, we have obtained some lessons learned about the dedicated SpaceWire usage. First, we report the problems and workarounds after making an NVDR architecture summary. Secondly, we report a Next-Generation NVDR (NVDR GEN2), which we are designing for future demands. NVDR GEN2 project is supposed to upgrade current NVDRs with several times higher data rate and larger capacity. But the current NVDR internal architecture which is composed of SpaceWire and UT54LVDS217/218s [3] [4] is not be able to handle larger band width. That is why we are considering introducing SpaceFibre. We report a configuration for a memory board using SpaceFibre.

II. NVDR ARCHITECTURE

We show NVDR specifications in the table I.

NVDR specification list		
Item	Value	
Capacity	1 Tbyte	
Input Data Rate	9.43 Gbps	
Output Data Rate	3.6 Gbps	
Mass	40.7 kg	
Power	Max. 100 W	
Input Interface	UT54LVDS218 RS422	
Output Interface	TLK2711 [5] RS422	
CMD/TLM Interface	MIL-STD-1553B RS422 SpW	

A block diagram of an NVDR is shown in Fig. 1. The architecture has following features:

- Payload data links and a SpaceWire control network are independent from each other to avoid data collisions.
- A SpaceWire control network is built between a control board and memory boards through a SpaceWire router. The SpaceWire router is implemented in the control board and the SpaceWire network goes through a passive SpaceWire Backplane.
- Input interface boards provide UT54LVDS218 inputs for high rate payloads and RS422 inputs for low rate ones. An output interface board outputs replay data to down-link modules through TLK2711.
- Selectable OBC interfaces are available: MIS-STD-1553B, RS422 or SpaceWire.
- Memory controllers can remove not only SEUs but also NAND flash functional errors. And NVDR's internal redundancy mitigates one failure.

NVDR software works in a 20 MHz FPGA based SoC which embeds a SPARC V8 processor as a single task. The software controls a file system following OBC commands and sends addresses to memory controllers through a SpaceWire network in the NVDR. The addresses designate locations where payload data should be recorded in NAND flash areas. After the software receives a start command for recording, input interfaces are set enable and memory controllers receive payload data from input interface boards through UT54LVDS217/218 links and write the data to in NAND flash memories. To replay data, an OBC selects replay files and sends the commands to the NVDR. Following the commands, the NVDR sets addresses to memory controllers and memory controllers read data from NAND flash memories. The replay data are transferred to the output interface board through UT54LVDS217/218 links. UT54LVDS217/218s are drove by 70 MHz clocks and achieve 1.4 Gbps data rate.

NVDR interfaces		
Item	Interface	
Input I/F	UT54LVDS218 20 ch RS422 8 ch	
Output I/F	TLK2711 3 ch	
OBC I/F	MIL-STD-1553B RS422 SpW	
Internal Payload Link	UT54LVDS217/218	
Internal Control Network	SpaceWire/SPI	

III. SPACEWIRE NETWORK AND PROTOCOL

In an NVDR, a SpaceWire network is built in a Backplane Board. The Backplane board is too large size: 540 mm x 374 mm x 294 mm to go through our reflow oven. Moreover, because the backplane board should be resizable according to missions, the mechanical redesigning cost for active



Fig. 1. NVDR block diagram

backplanes becomes higher. That is why we chose a passive backplane. As a result, a SpaceWire router is implemented in a FPGA on the control board. In the NVDR, an MPU exists as only one initiator node. We introduced a simple SpaceWire router with only one initiator node. This reduced implemented resource in the router FPGA. This location leads to a pin-neck problem in the control board. The control board must be connected to 12 memory controllers including redundancy. Therefore 80 pins are necessary for SpaceWire signals at the control board connector to the backplane board. To save the number, we introduced LVTTL instead of LVDS for SpaceWire. Consequently, we are compelled to restrict the SpaceWire link rate to 10 MHz. But the link rate can fit to required bandwidth for the NVDR. To control high input payload data, software control overheads for handling SpaceWire packets becomes a more serious problem rather than the link rate.

To resolve the problem, the NVDR uses a dedicated protocol which is based on RMAP. We call the protocol NPTP (NVDR packet transfer protocol).

A. NPTP protocol

NPTP protocol is based on RMAP protocol, but it is simplified to reduce software handling overheads. We omitted RMAP headers related to error detection and a reply packets for a write command acknowledge. Because memory boards require new addresses as soon as possible after transferring



data to NAND flash memories, we introduce a fixed-length

Target Address	PID	Source Address	UID	DATA 64 Byte	EOP
F ' 0 I		D 1 . I	7 .		

Fig. 3. Interrupting Packet Format

interrupting packet which memory boards can send at any time to notify the demands. NPTP transactions are shown in Fig. 2. The interrupting packet format is indicated in Fig. 3.

B. SpaceWire to SPI

To reduce backplane connector pins, we converted SpaceWire to SPI (Serial Peripheral Interface) only between the control board and interface boards. Unlike memory boards, interface boards do not require frequent accesses, so low rate SPI is enough for the control. And interface boards do not notify anything to the control board by sending interrupting packets, so we required NPTP reply packets to verify the operation. The SPI is composed of CLK, MISO and MOSI which are common in all interface boards in addition to individual CS_L signals. LVTLL SpaceWires would need 20 pins for 5 interface boards, but the SPI requires only 8 pins. The SPI clock is 1 MHz. The common CLK sampled by each FPGA clock on interface boards suppresses noises at falling or rising edges.

C. Simple Router

In the NVDR SpaceWire network, the control board is only one initiator except each memory board can send interrupting packets spontaneously. A 1-to-N router can fit to this network configuration and a 1-to-N router can reduce resources when implemented in a FPGA.

IV. LESSONS AND LEARNED

From performance evaluation of an NVDR, we obtained some lessons and learned. First, software overheads degraded record and replay performance when an NVDR software sends and receives SpaceWire packets for replay and record addresses. Second, wrong implementation of an interrupting packet generator and an NPTP protocol target caused deadlocks in the software.

A. Degradation by software overhead

NVDR software must restart DMA controllers every time receiving and sending a SpaceWire EOP. The higher an NVDR input/output data rate becomes, the more fixed-length interrupting packets are generated from memory controllers. Therefore, the software control overhead for DMAs degrades NVDR input/output data rate performance. To reduce EOP receipts, we allowed memory controllers to make a chain of interrupting packets. A chain of interrupting packets consists of some fixed-length interrupting packets and one EOP. To make a long chain, memory controllers should buffer interrupting packets in some time.

B. Software deadlocks

Because of wrong implementation, memory controllers cannot process NPTP Read commands nor NPTP write commands while trying to send an interrupting packet. Therefore, NVDR software, which runs as a single task, could not receive the interrupting packet while trying to send an NPTP Read command or an NPTP write commands. To resolve this dead lock, we added a deep FIFO buffer in the 1to-N SpaceWire router. This buffer is designed to retrieve all the interrupting packets rapidly from memory controllers when NVDR software tries to send NPTP commands.

V. NVDR GENRATION2

For future earth observation satellite's demands, we launched a new project to upgrade current NVDRs to next generation NVDRs (NVDR GEN2). We predict that 5 times faster input/output data rate and 2 times larger capacity will be required for future demands. Now we are redesigning memory controllers and interlinks. For an internal link, the required data rate reaches 12.5 Gbps bandwidth. Therefore, it is difficult for current links with UT54LVDS217/218s to meet this data rate. So, we are considering adopting SpaceFibre. In this chapter, we look over the whole architecture. Secondly, we consider a SpaceFibre configuration for one memory board.



Fig. 4. NVDR GEN2 block diagram

VI. NVDR GENERATION2 ARCHITECTURE

Fig. 4 shows a block diagram for an NVDR GEN2. We are considering upgrading the current processor to a higher performance type. For example, GR740 [6] is one candidate because we consider that multi-core processors fit for handling memory controllers. And SpaceWire link rate should be boosted to more than 100 MHz to accommodate more interrupting packets. In the current NVDRs, RTAX2000s are used as memory controllers, but we assume that Microsemi's RTG4s [7] are needed for SerDes interlinks and high-speed memory control.

TABLE III.	SPACEFIBRE CONFIGURATION FOR A MEMORY BOARD

Item	SpaceFibre
Development Status	Developed ^a
Links for record	3
Lanes for record	6 receiver lanes 2 transmitter lanes
Links for replay	2 or 3
Lanes for replay	2 receiver lanes Transmitter lanes are TBD

^{a.} Multi-lane function is under development.

VII. USAGE OF SPACEFIBRE AND SPACEWIRE

To achieve 12.5 Gbps data rate for record, five transmitter lanes are needed in one memory board if a single lane is 2.5 Gbps. SpaceFibre, which we also developed [8], uses at least one opposite directional lane because of its full-duplex protocol. Moreover, Microsemi's report [9] shows RTG4's SerDes is not complete upset-free and sometimes needs reset operation for its recovery. Because the recovery takes sub-millisecond duration if the recovery process happens, SpaceFibre must keep retrying sending data until the SerDes recovers. For example, considering 1 ms duration for the recovery, SpaceFibre transmitter side must accumulate 1.5 Mbyte data with 12.5 Gbps data rate. Therefore, SpaceFibre transmitter side requires an external buffer in addition to the retry buffer. To mitigate the recovery process with multi-lane, one more lane must be added to each transmitter and receiver side. It means that total 6 transmitter lanes and 2 receiver lanes in a memory board are needed for one SpaceFibre link. For redundancy, we require a memory board to provide 3 SpaceFibre links for record. And one RTG4 has total 24 lanes for each transmitter and receiver. Therefore residual 6 lanes of 24 receiver lanes can be allocated to SpaceFibre links for replay. Similarly, a memory board must have at least two lanes for sending ACK/NACKs to an output board. It means that one memory board can have at most three SpaceFibre links which are connected to output interface boards.

From the view point of NVDR GEN2 control network, SpaceFibre provides a possibility of deleting all the SpaceWire network except through one link between the control board and an output interface board. The control board can control all the other boards over SpaceFibre links because SpaceFibre can transfer deterministic data along with high rate data.

We summarize a SpaceFibre configuration for one memory board in the table III. To meet the configuration, we will continue to develop SpaceFibre using existing design resources.

VIII. CONCLUSION

In this study, we showed how SpaceWire is applied in an NVDR. SpaceWire contributes effectively to the record and replay performance with some lessons learned. Second, we explained NVDR GEN2 architecture and pointed out how many lanes are needed for SpaceFibre in one memory board when considering required data rate and RTG4's PLL upsets. Moreover, we estimated the retry buffer size. In 2018, we will start developing proto-type memory boards and evaluate the performance.

ACKNOWLEDGMENT

The author thanks the many people who assisted with the concept and offered support all along the way.

REFERENCES

- Karl F Strauss, "Memory Technologies and Data Recorder Design," IEEE Aerospace conference. Big Sky, Montana, p. 1-15, 2009
- [2] T.Sasaki, "Application of SpaceWire to Non-Volatile Data Recorder," International SpaceWire conference 2013, Gethenburg, Sweden, June 2013
- [3] Cobham plc., "UT54LVDS217 Serializer," http://ams.aeroflex.com/pagesproduct/datasheets/lvdsserializer.p df.
- [4] Cobham plc, "UT54LVDS218 Deserializer," http://ams.aeroflex.com/pagesproduct/datasheets/lvdsdeserialize r.pdf
- [5] Texas Instruments Inc,
 "TLK2711-SP 1.6-Gbps to 2.5-Gbps Class V Transceiver," http://www.ti.com/lit/ds/symlink/tlk2711-sp.pdf
- [6] Cobham Gaisler AB,
 GR740 Quad-Core LEON4 SPARC V8 Processor," http://gaisler.com/index.php/products/components/gr740
- [7] Microsemi Corporation, "PB0051 Product Brief RTG4 FPGAs," https://www.microsemi.com/documentportal/doc_download/134430-pb0051-rtg4-fpgas-product-brief.
- [8] Takahiko Masuzaki, Minoru Nakamura, Tetsuro Kato, Yasunori Ido, Toru Sasaki, "IMPLEMENTATION AND INTEROPERABILITY TESTS OF SPACEFIBRE," International SpaceWire conference 2013, Gethenburg, Sweden, June 2013
- [9] Microsemi Corporation, "RTG4 PLL SEE Test Results July 10 2017,"

https://www.microsemi.com/documentportal/doc_download/137654-rtg4-pll-and-internal-oscillatorsee-report

Standardized High Performance SpaceVPX Modules Leverage SpaceWire as an Internal/External Control Fabric

On-Board Equipment and Software, Short Paper

Joseph Marshall BAE Systems Manassas, Virginia, USA Joe.Marshall@baesystems.com

Abstract— This paper describes BAE Systems' SpaceVPX architecture, the various SpaceVPX modules and daughter cards under development with special emphasis on the SpaceWire power conversion module and the 1553-SpaceWire single board computer daughter card. Interconnects between modules are highlighted with special emphasis on SpaceWire ports, routers and backplane routings. Performance of the modules and their interconnects and support software are summarized.

Index Terms—SpaceWire, Networking, Spacecraft Electronics, SpaceVPX, I2C, Power Conversion, PCI, RapidIO, SpaceVPX Lite, Single Board Computer, RADNET Components, RMAP

I. INTRODUCTION

New and future spaceborne systems require additional onboard processing and much greater interface connectivity. Many efforts worldwide are starting to address these needs. SpaceVPX, an ANSI/VITA standard released in 2015, was created to provide the structure and definition for interoperable modules to meet these needs. As shown at MAPLD 2017[1], multiple organizations are utilizing the standard and its companion, SpaceVPXLite, in their new module development. SpaceVPX defines a multi-layer set of fabrics using SERDES, LVDS and LVCMOS devices to provide standardized interconnects in a scalable and fault tolerant way. SpaceWire is setup as both a control plane for command and data handling throughout the box as well as a medium speed data plane [2].

Building on previous SpaceWire network elements, BAE Systems is continuing to develop a set of silicon application specific standard products (ASSP) to provide power efficient general purpose building blocks for the creation of scalable SpaceVPX modules across these fabrics. These building blocks are key to a new family of SpaceVPX processing and network modules being developed for a wide variety of space applications. These include a RAD750® Single Board Computer, a RAD5545[™] Single Board Computer, a Virtex-5 based Reconfigurable Computer Module, a RapidIO Switch Module, multiple types of standalone function modules focused on memory and/or unique I/O and a SpaceWirecontrolled Power Conversion module. All of these modules contain at least dual redundant SpaceWire ports to be utilized by the System Controller for command and data handling or between modules for medium speed data transfers. BAE Systems is also creating a 1553-SpaceWire daughter card for the RAD5545 SBC which routes 1553 and SpaceWire links to module top card connectors for extending the control, data handling and data transmission outside of the SpaceVPX backplane.

II. SPACEVPX ARCHITECTURE

ANSI/VITA 78.00, also known as SpaceVPX[™] [3][4][5], (a space version of OpenVPX[6]) was ratified in 2015 and is used as the basis of most new BAE Systems spacecraft electronics modules. Fig. 1 illustrates a notional spacecraft architecture following the SpaceVPX standard. Redundancy of modules is not shown.

Payload functions are represented by the upper dashed box. Spacecraft bus functions are shown by the lower dashed box and show connectivity from older heritage devices. Sensors and remote spacecraft electronics are shown at the periphery. Running horizontally across Fig. 1 are the three main fabrics defined in SpaceVPX. RapidIO[7] is used as the data plane for passing data between modules at rates up to 20 Gbps. SpaceWire is the used as the control plane with usage envisioned both for medium speed data transfers (10 to 500 Mbps) and more importantly for command and data handling of the electronics. I2C is used as the main redundancy control and basic telemetry interface for basic platform operation and passes information at rates up to 400 Kbps. It is supplemented by Utility Plane resets, clocks, test and power interfaces.

SpaceWire is a key interface on the modules throughout the architecture. In the upper payload box, payload or peripheral modules typically have a minimum two SpaceWire ports to attach to redundant system controllers. Each of the two Single Board Computers (SBC) shown in Fig. 1 may be utilized as SpaceVPX System Controllers. In this mode, they provide SpaceWire links to all other logic modules in the box as well as similar Utility Plane connections either directly or through Utility Plane Control switch functions on other modules. As such, these SBCs contain SpaceWire routers to enable the connection of multiple links to each system controller.

The Bus architecture block also shows this approach at a simpler level where the SBC connects to various sensors and instruments via SpaceWire but talks to heritage electronics modules over the older CompactPCI[8] spacecraft bussed data interface.



Fig. 1 Spacecraft Electronics SpaceVPX Architecture

III. SPACEWIRE ON SPACEVPX MODULES & COMPONENTS

Fig. 1 shows a variety of functional SpaceVPX modules typically utilized to create a spacecraft electronics payload or box. The value of SpaceVPX is realized in the standardized interfaces - though BAE Systems modules are shown, each may be mixed and matched with other industry SpaceVPX modules utilizing the same slot and module profiles.

There are two SBCs shown in Fig. 1. The first (leftmiddle)[9] is built around the RAD750[®] CPU[10], a processor capable of 250-500 Dhrystone MIPS, volatile SRAM memory, non-volatile Flash memory, and the RADNET[™] SpW RB4 [11][12] (4 port remote bridge) application specific standard product (ASSP), a bridge device for the RAD750 CPU connecting it to memory, the CompactPCI bus, a four port SpaceWire router, an embedded 16 DMIP microcontroller (EMC) [13][14] and several other interfaces including MIL-STD-1553B. SpaceWire links operating up to 320 MHz may either utilize the RB4's four port SpaceWire router or directly connect to DMA engines in the RB4. The RB4 supports the SpaceWire Remote Memory Access Protocol (RMAP) in hardware and may handle any other SpaceWire protocol by passing packets to its internal EMC.

The second SBC (top-middle of Fig 1.) is assembled around the RAD5545TM system on a chip (SoC)[15], a 5.2

GIPS quad core processor with four 5 GHz RapidIO ports, pluggable volatile DDR3 memory, 1 GB non-volatile Flash memory, and sixteen 400 MHz SpaceWire ports. Up to eight of these ports may be used directly without a router or all sixteen may be used with the on-chip 16 port router. The RAD5545 SoC directly supports RMAP and also has a 20 DMIPS EMC that may be used for other SpaceWire protocols. Twelve of the SpaceWire ports are wired to the SpaceVPX backplane for use in either payload or system controller applications. The other four are wired to a daughter card connector providing the ability to personalize this module for unique external interfaces. Fig. 2 is a diagram of the RAD5545 SBC [16].

A 1553-SpaceWire daughter card (shown in Fig. 2) is under development that adds a dual redundant MIL-STD-1553 interface to RAD5545 SBC through a PCI interface and buffers SpaceWire ports for external connection. Top of card connectors are provided for the 1553 A and B connectors and for four SpaceWire links. Due to space limitations on the header of the daughter card, the four SpaceWire links exit the module through a 37 pin connector – 9 pins for each interface and a single chassis ground. A matching cable harness will break out each SpaceWire port to a separate SpaceWire cable. This allows the RAD5545 SBC to control both the interfaces in the SpaceVPX chassis and to interface to up to four devices external to the chassis.



Fig. 2 RAD5545 SBC with SpaceWire-1553 Daughter Card

Returning to Fig. 1, there is one other processing module, the Reconfigurable Computer Module (RCM) [17] shown in the top right of the payload. This module contains reprogrammable field programmable gate array (FPGA) devices such as the Xilinx Virtex-5 FPGA [18] supported by volatile DDR2 memory, flash non-volatile memory to hold the gateware configurations and an I/O daughter card connector for each FPGA as well as a utility device to load gateware and monitor the FPGA for radiation effects. The RCM is used for signal processing and other digital conversion applications. SpaceWire is used as the control plane interface and provides the system controller with access for commands and telemetry as well as configuration loading. The support device may be as simple as an RTAX fuse-based FPGA (150 MHz SpaceWire links), RTG4 flash-based FPGA (200 MHz SpaceWire links) or as capable as the RADNET SRIO-EP ASSP [19] (400 MHz SpaceWire links).

The RADNET SRIO-EP is a RapidIO Endpoint with considerable controller, memory and interface capabilities. Besides a dual redundant RapidIO Endpoint interface, it has two dual-redundant XAUI interfaces, two DDR3 volatile memory interfaces, 1553 interface, four high performance (100 DMIP) microcontrollers, a 16 DMIP EMC, and a variety of other interfaces including I2C and four ports of 400 MHz SpaceWire that may be used directly or through the on-chip four port router. Besides its potential use on an RCM, this device is used as the SpaceVPX all- fabric interface for the Recorder [20], Communications Interface and Sensor Interface modules in Fig. 1.

The final logic module shown in the payload box of Fig. 1 is a network switch [21]. This provides switching of the

RapidIO interfaces in a box when a simple mesh between modules (typically five or less) is not sufficient or when several data plane interfaces come from external sources and need to be interfaced to the rest of the system. Switching is provided by the 48-lane 18 port RADNET 1848-PS and the 1616-XP cross point switch ASSPs [22]. Once again an RTAX or RTG4 is called upon for the utility plane interface as well as a 100-200 MHz pair of SpaceWire links. These links are typically used for control and telemetry, especially when the data plane is being configured or reconfigured.

IV. SPACEWIRE-CONTROLLED POWER SUPPLY

SpaceVPX spells out two secondary power sources for the fault tolerance of the box but leaves definitions of power supply modules mostly out of its purview relying on VITA 62. This standard has not moved in a very usable direction for SpaceVPX systems. Thus the creation of the new SpaceVPXLite (VITA 78.1) and a revision of SpaceVPX are both underway and will provide some interoperable power supply definitions. In creating multiple SpaceVPX boxes and assemblies, BAE Systems has settled on a Space-VPX compatible module approach that it plans to take forward for consideration in the SpaceVPX revision.

Fig. 3 shows a block diagram of the Power Converter Module. It contains power converter and EMI block devices to convert the primary power to the secondary voltages needed on the SpaceVPX utility plane (+3.3V, +5V, +12V, -12V). These are controlled by the RADNET SpaceWire Endpoint (SpW-EP) [23][24][25] which like the RB-4 above contains a 16 DMIP EMC, internal SRAM and several interfaces including external memory, I2C, discretes and a 320 MHz dual redundant SpaceWire port. The endpoint is able to boot from the onmodule non-volatile memory as soon as the 3.3V converter on board reaches stable power. It can then monitor appropriate telemetry as it enables other converters for the rest of the modules in the SpaceVPX box. Once it powers the System Controller, the System Controller may utilize the SpaceWire RMAP capability to access Power Converter registers and take over the monitoring function for the box with the EMC now providing watchdog functions. This generic setup with a small controller with both utility and control plane interfaces enables sharing of control between single string implementations as well as full dual setups.



Fig. 3 SpaceWire-controlled Power Converter Module Block Diagram

The Power Converter Module fits in the 6U SpaceVPX footprint. It defines a new combination of wafers by repeating the P0 connector layout eleven times within the 6U width. These are numbered P0 to P10. The module slot definition showing these eleven wafer sections is shown in Fig. 4.

Each of the modules uses the top two wafers for power outputs isolated from the signals by a ground wafer in the third position. The next three wafers contain single-ended signals and the final two wafers contain differential signals. As such, the utility plane connections match the standard SpaceVPX connections in P0. P5 and P6 are used for the control plane SpaceWire connections.

SpaceVPX always envisioned that all telemetry signals would be captured on the module and made available through the utility or control planes to external modules. Due to available devices, real estate and other considerations, this has not yet fully played out. Thus, our Power Controller module has the ability to collect and condition telemetry signals from other internal modules. These are then readable either by the EMC in the SpaceWire Endpoint or by an external intelligence using RMAP commands over the SpaceWire control plane. It is projected that someday this may not be needed but this serves the need today.

SpaceWire as a power supply interface is a departure from the terrestrial OpenVPX VITA 62 standard. Another is not handling the primary power on the backplane. The Power Converter receives and returns primary power and responds to any primary commands using its top of card connectors. This enables better isolation on the power converter module.

Software support for controlling or using each SpaceVPX module is being developed as VxWorks drivers and board support in conjunction with each module's development and the systems controller SBC targeted to control that module.

V. SUMMARY

Demand for growing requirements onboard processing is being met by a family of SpaceVPX modules created or under development by BAE Systems. These modules use a tiered set of electronics fabrics to interconnect the modules and control, sense and pass data. SpaceWire is one of the three interfaces and is key to systems controlling and moving configurations, telemetry or medium speed data between modules.

BAE Systems SpaceVPX modules utilize a set of interface and processing building blocks providing SpaceWire links and routers from 1 to 16 ports with frequencies of 10 to 400 MHz. Our latest development is a 1553-SpaceWire daughter card to personalize the I/O on a RAD5545 SBC.



Fig. 4 Power Converter Module Slot Definition

We have created a Power Converter Module compatible with yet filling a gap in the SpaceVPX standard that relies on SpaceWire for connections to the logic modules.

On the horizon for SpaceVPX includes SpaceFibre as an alternate data plane. SpaceFibre's close connections to SpaceWire should keep SpaceWire key in future SpaceVPX.

VI. REFERENCES

- [1] MAPLD SpaceVPX Session, SEE/MAPLD Conference, San Diego, CA, May 2017.
- [2] ECSS Standard ECSS-E-ST-50-12C, "SpaceWire, Links, Nodes, Routers and Networks", Issue 1, European Cooperation for Space Data Standardization, July 2008.
- [3] "SpaceVPX Standard", ANSI/VITA 78.00-2015, www.vita.com
- [4] Collier, Charles Patrick, et al., "Next Generation Space Interconnect Standard (NGSIS): A Modular Open Standards Approach for High Performance Interconnects for Space", Proceedings of 2015 IEEE Aerospace Conference, Big Sky MT USA, March 2015
- [5] P. Collier, J. Marshall, R. Berger, M. Enoch, S. Goedeke, "Next Generation Space Interconnect Standard (NGSIS): A Modular Open Standards Approach for High Performance Interconnects for Space", AIAA 8 Reinventing Space 2013 Conference Proceedings, Los Angeles CA USA, September 2013.

- [6] "OpenVPX System Specification", ANSI/VITA 65-2010 (R2012), 2012, www.vita.org
- [7] RapidIO Interconnect Specification 3.1, www.rapidio.org, September 2014
- [8] Compact-PCI Base Specification, PICMG 2.0 Rev 3.0, October 1, 1999. www.picmg.org
- [9] J. Marshall, R. Berger and L. Assadzadeh "SpaceWire Fabric Used to Control Family of Standardized High Performance SpaceVPX Modules", Proceedings of the 2016 International SpaceWire Conference, Yokohama, Japan, October 2016.
- [10] R. Berger, et al, "The RAD750 A Radiation Hardened PowerPC Processor for High Performance Spaceborne Applications", IEEE Aerospace Conference 2001, Big Sky MT, USA, March 2001
- [11] J. Marshall and R. Berger, "High Performance Network Components for Scalable Spaceborne Processing Needs", Proceedings of the 2016 International SpaceWire Conference, Yokohama, Japan, October 2016.
- [12] Marshall, J. R., "Evolution and Application of System on a Chip SpaceWire Components for Spaceborne Missions", 2nd International SpaceWire Conference, Nara, Japan, 2008.
- [13] J. Marshall, D. Stanley, J. Robertson, "Matching Processor Performance to Mission Application Needs", Infotech@Aerospace 2011 Conference Proceedings, St. Louis MO, USA, March 2011.
- [14] J. Marshall and J. Robertson, "An Embedded Microcontroller for Spacecraft Applications", IEEE Aerospace Conference 2006, Big Sky MT, USA, March 2006
- [15] R. Berger, et. al., "Quad-Core Radiation-Hardened System-on-Chip Power Architecture Processor", Proceedings of the IEEE Aerospace 2015 Conference, Big Sky MT USA, March 2015.
- [16] Marshall, J., et. al., "Maturing SpaceVPX High Performance Computing Modules for Next Generation Onboard Processing", Proceeding of GOMACTech 2017 Conference. Reno, NV, March, 2017.
- [17] Marshall, J., et. al., "SpaceVPX High Performance Computing Modules for Next Generation Onboard Processing", Proceedings of GOMACTech 2016 Conference, Orlando, FL, March 2016.
- [18] Radiation-Hardened, "Space Grade Virtex-5QV Family Overview", Xilinx DS192 version 1.3, March 2012.
- [19] D. Rickard, et. al., "On-Board Networks with Radiation Hardened 45nm SOI Standard Components", Proceedings of the IEEE Aerospace 2015 Conference, Big Sky MT, March 2015.
- [20] D. Matthes, et. al., "Maturing SpaceVPX High Performance Computing Modules for Next Generation Onboard Processing, Proceedings of GOMACTech 2018 Conference, Miami, FL, March 2018.
- [21] Kelly, A., et. al., "Single Event Effects Characterization of BAE Systems RADNET 1848-PS RapidIO Packet Switch", NSREC Conference Proceedings, New Orleans, LA, July, 2017.
- [22] Rickard, et al, "On-Board Networks with Radiation-Hardened 45nm SOI Standard Components", IEEE Aerospace Conference 2015, Big Sky MT, March 2015
- [23] [18] J. Marshall, "Standardized SpaceWire Solutions for Next Generation Systems", Proceedings of the 2014 International SpaceWire Conference, Athens, Greece, September 2014.
- [24] J. Marshall and R. Berger, "A One Chip Hardened Solution for High Speed SpaceWire System Implementations", International SpaceWire Conference 2007, October 2007
- [25] J. Marshall, S. Santee, M. Hanley, J. Robertson, D. Stanley, "Leveraging SpaceWire Networking Prototyping to Create Flexible SpaceWire Components and Support Software", International SpaceWire Conference 2011, San Antonio TX, USA, November 2011.
- All figures are Copyright 2018 BAE Systems used with permission

Abstracted SpaceWire Interface for Deep Space Radio

PAPER NOT PUBLISHED

Networks & Protocols (Short)
Using SpaceWire Time-codes for Global Synchronization of PLL-Based Local Clocks

SpaceWire Networks and Protocols, Short Paper

Thomas Bahls and Markus Bihler Institute of Robotics and Mechatronics German Aerospace Center (DLR) Münchnerstr. 20 82234 Weßling, Germany Thomas.Bahls@dlr.de

Abstract—In a control application all sensors and actuators have to be able to perform their functions in a specific time to guarantee the desired control cycle. To achieve an exact actual system state a synchronization of the different sensor measurements is necessary. This guarantees that all acquired values are in a very small period of time and not spread over the whole control cycle. Otherwise, a noiser system state would result. The simplest way to synchronize those measurements is with cyclic triggering. This can be either an external request via communication or generated locally by a timer. Both approaches work well in a monolithic system approach with a central unit due to the existence of a common clock domain. However, a distributed system consists of multiple monolithic systems connected via communication, each with a separate clock domain. Due to the communication delay from unit to unit, and the differences between the local clocks, a synchronization according to the presented approaches is only possible by introducing a global clock domain.

With time-codes SpaceWire already includes a mechanism to distribute time over a network. However, the communication delay regarding the time-code distribution is not considered. In systems consisting of large networks which have to be synchronized in very small cycles (e.g. high dynamic robots see [1]) this aspect is not negligible. This approach introduces a global time distribution by synchronization and runtime compensation. The synchronization is based on digital phase locked loops (PLLs) which align the units' local time to the occurence of SpaceWire time-codes. The runtime compensation of the timecode distribution in the network is based on statistical evaluation.

I. INTRODUCTION

In a control cycle of a system the desired data for the actuators is computed by a control model based on the actual data (sensor values and internal states) of the system. The actual data represent the system state ideally at one specific point in time. The aim is to perform the acquisition of all actual data only in a small time window as close as possible to one specific point in time. Otherwise the different values are not comparable and a more noisy system state is the result. Hence a synchronization has to be introduced to guarantee data acquisition from different sources simultaneously.

Providing synchronized data cyclic from different sources results in a time triggered system [2]. This is why the introduction of a global system time as a reference is necessary. In a monolithic system approach with a central unit the realisation of a global system time is easy since all data sources are directly connected to it. In this case, the global system time is equal to the time of the central unit. In contrast, a distributed system consists of various spatially separated units connected via communication, each with its own local time. Here, the global system time has to be provided via communication by synchronizing the local time to a global system time. In robotics the strict real-time requirements (main control loops $\geq 3KHz$) and the size of the network (see Fig. 1) often prohibit negligence of the time distribution runtimes in the network.



Fig. 1. SpaceWire network topology of a tpyical robot. Squares match to nodes, octagons to routing switches, and rectangles to exchange levels

The time distribution in the network depends on delay and jitter. The aim is to neutralize both effects. SpaceWire supports with its time-code mechanism an approach for distributing time in a SpaceWire network [3]. However, a local time implementation as well as synchronization to a global time in combination with compensation of the communication delays and the effect of jitter are not yet available. Other standards such as EtherCat [4] or Flexray [5] already integrate such approaches, or use TTP [2] and similar methods to expand existing standards by a time synchronisation (e.g. TTE [6] or TTCAN [7]).

This paper descirbes a generic local time approach based on a digital PLL which cyclically adjusts an numerically controlled oscillator (NCO) via SpaceWire time-codes. The communication delay is also considered via an individual offset for each local time. In section II the chosen approach is



Fig. 2. Digital PLL which cyclically adjusts the NCO's internal increment regarding the actual phase difference.

presented in detail. Section III describes the test hardware and the performed measurements. The paper concludes in section IV and gives an outlook in section V.

II. CONCEPT

The presented approach assumes an arbitrary SpaceWire network topology with an arbitrary number of nodes and a single time-code generation source in the network. The time-codes which are used to synchronize the local time in the different nodes are generated cyclically with the frequency f_{sync} . The core elements of the local time implementation is a digital PLL which adjusts the NCO's output frequency regarding the received synchronization signal. Fig. 2 shows the structure of the digital PLL in combination with the NCO schematically.

The NCO's output frequency is defined as follows

$$f_{NCO} = n \cdot f_{sunc} \qquad n \in \mathbb{N}_+ \tag{1}$$

The NCO itself runs with in the local system frequency f_{sys} and has an accumulator width of m bits. m is defined by

$$m \gg floor(\frac{\ln(\frac{f_{sys}}{f_{NCO}})}{\ln 2})$$
 (2)

It is recommended to choose an adequate accumulator width since the frequency accuracy of f_{NCO} rises with m.

At start up, each NCO waits until an initial time-code is received (which equals a rising edge on f_{sync}). After that, the NCO starts to increment its accumulator in each f_{sys} cycle (see top of NCO block of Fig. 2) by the standard increment calculated as follows

$$i_{standard} = \frac{2^m \cdot f_{NCO}}{f_{sys}} \tag{3}$$

With the next rising edge on f_{sync} the phase difference against the last rising edge on f_{NCO} is calculated. The resulting phase difference φ is fed into a sample and hold block (see S/H block in Fig. 2) which samples on the rising edge of f_{sync} . With the registered phase difference φ_s the value is available until the next sync event, since the adaption of the NCO is performed smooth over the next *n* cycles of f_{NCO} and not in a single cycle of f_{sys} . The additional increment for correction which has to be added to the $i_{standard}$ in every cycle of f_{sys} is calculated with

$$i_{correct} = \varphi_s \cdot k \qquad k = \frac{f_{sync}}{f_{sys}}$$
(4)

Depending on the application and the cyclic occurrence of the f_{sync} pulses the adaption can be preformed even smoother over several *n* cycles of f_{NCO} simply by multiplying *k* with $\frac{1}{l}$ with $l \in \mathbb{N}_+ \setminus 1$.

The digital PLL is considered as locked if the phase difference undercuts a specific threshold which equals the expected phase difference occurred due to communication jitter in the time-code distribution. As long as this condition is fulfilled f_{NCO} is cyclically aligned to the received f_{sync} pulses. Exceeding the threshold leads to a restart of the procedure.

With the digital PLL the NCO's f_{NCO} can be aligned to the globally distributed time-code pulses which are locally represented by f_{sync} pulses. However, due to different communication delays regarding the position in the network the NCOs in the different nodes are not aligned among each other. Therefore, a second derivated frequency domain $f_{NCO'}$ is introduced. $f_{NCO'}$ is phase shifted to f_{NCO} by adding a constant phase φ_d which equals the individual communication delay of every node from the time-code generation source. The former presented alignment of f_{NCO} stays unchanged. With the compensation of the individual communication delay all $f_{NCO'}$ in each node is aligned to each other. By synchronizing all data acquisition procedures in each node to their local $f_{NCO'}$ all values are in a small time window very close to a single specific point in time.



Fig. 3. Timestamp in global time representation

 $f_{NCO'}$ is also used to generate unique global timestamps in the SpaceWire network. By additionally counting the wrapping of $f_{NCO'}$, timestamps are provided containing the $f_{NCO'}$ counter as LSBs and the wrapping counter as MSBs (see Fig. 3). The timestamp is used e.g. for ordering of the data or to determine the roundtrip time. Depending on the application the bit width of the wrapping counter has to be chosen to guarantee an unique timestamp during the runtime of the system.

III. RESULTS

The presented approach is tested and evaluated based on SpaceWire ready joint electronics used in the medical robot DLR Miro [8]. It provides two exchange level implementations in accordance to [9] for interconnection in combination with a Gbps capable physical level in accordance to [10]. Both as well as the remaining SpaceWire stack are implemented in an FPGA. Four of this joint electronics are chained together as depicted in Fig. 4.



Fig. 4. Test setup for evaluation.

The electronics are connected via their exchange level implementations. All electronics operate with a system clock of $f_{sys} = 62.5MHz$. Electronic 1 is the time-code source which generates time-codes in $f_{sync} = 3kHz$. The electronics 2 to 4 implement the former presented local time approach with $f_{NCO} = 96kHz$ and an accumulator width of m = 32.

Initially the variations of the used electronics are measured. Therefore, each used electronic is directly connected with the time-code source and a window of 5000 time-codes is recorded for each. The mean duration varies in a range from 188.59 ns to 193.75 ns, the jitter from 37.3 ns to 38.6 ns, and the standard deviation from 5.40 ns to 5.47 ns. This shows the tolerance of the of the identically assembled electronics with the same firmware. Fig. 5 shows one histogram and its statistical evaluation exemplarily. Due to the determinism of the free running synchronization approach based on the two flop synchronizer scheme according to [11] the distribution deviates from a pure Gaussian.



Fig. 5. Histogram of the communication delay between electronic 1 and 2.

The next measurement comprises additionally the duration of the time-codes to chain position 3 and 4 (see Fig. 4). Again windows of 5000 time-codes are recorded for the statistical evaluation of the durations. Here, it is recognizable that the communication delay follows more and more a Gaussian distribution with increasing number of exchange level implementation to pass by the time-codes. This is obvious since the synchronizations in the exchange level are independent from each other. They are running on different system clocks and have been started in different points in time. This leads to an averaging of the time-code communication delay distribution with an increasing number of exchange levels passed by the time-codes. Fig. 6 shows the communication delay distribution in the last electronic in the chain. Here, the Gaussian distribution is clearly recognizable.



Fig. 6. Histogram of the communication delay between electronic 1 and 4.

The measured delays are now used in the PLL approach as local φ_d to generate the local $f_{NCO'}$ in each individual electronic. In general, the PLL leads to a reduction of the jitter of about 8%. The determinism of the free running synchronization approach in the exchange level 1 is also recognizable with the activated PLL due to its deviation from a Gaussian distribution. This is also observable in the mean communication delay, which is reduced by 12 ns compared to the measured value without activated PLL. All consecutive communication delays of the exchange level, which follow a Gaussian distribution, only vary within the previously presented tolerances. By taking into account the reduced delay, due to the determinism of the synchronization in the first exchange level, in all local φ_d the mean accuracy of all local $f_{NCO'}$ are in the resolution off f_{sys} . The effect of the presented approach becomes clear by comparing the receipt of the time-codes in each electronic (see Fig. 7) and the adjusted local $f_{NCO'}$ pulses (see Fig. 8).

IV. CONCLUSION

An approach for synchronizing the local time in spatially distributed units via global synchronization pulses has been presented on basis of SpaceWire. Nevertheless it is not limited to SpaceWire since dedicated synchronization pulses via communication can be provided in arbitrary communication



Fig. 7. Snapshot of the received time-codes in the in the electronics 2 to 4. First time-code generation followed by the reception in 2 to 4 respectively.



Fig. 8. Snapshot of the $f_{NCO'}$ pulses by use of PLL and communication delay compensation. First time-code generation followed by $f_{NCO'}$ pulses in 2 to 4 respectively.

systems. However, SpaceWire with its time-code approach provides already a mechanism for distribution of high priority messages in the complete network which makes the development of special protocols unnecessary. The approach is based on an NCO which is adjusted by a PLL according to received SpaceWire time-codes.

The results of the measurements show, that the chosen approach in combination with communication delay compensation was able to provide a mean resolution in the range of the local system clock over the chosen network topology. The use of the PLL also leads to a reduction of the jitter. According to the statistical evaluation of the received time-codes the experiments showed that the delay compensation in the first exchange level implementation, which follows the time-code source, have to be treated separately since it differs noticeably from a pure Gaussian distribution. The following exchange level implementations are in the range of the measured individual tolerance of the units. This shows the scalability of the approach for the usage in arbitray network sizes and topologies. The synchronization of all local times also enables the implementation of a global time representation. A format was presented which enables unique comparable timestamps in the network to allow e.g post ordering of the data or determination of roundtrip time.

V. OUTLOOK

The presented approach of globally synchronized local time enables the introduction of a fully synchronous communication approach with fixed timeslots, which also considers the communication delays. By reserving one timeslot exclusively for the distribution of the SpaceWire time-codes and changing the activation of the synchronizer scheme in the exchange level implementation from "free running" to "on demand" the jitter can be further reduced significantly. This results in higher absolute accuracy.

The presented approach uses pre-measured communication delays. It also has been contemplated to measure the communication delay at startup of the system. This also enables to take changes in the network topology into account, as well as the use of exchange levels of different speeds (e.g. 1 Gbps and 100 Mbps). Another advantage would be that each individual communication delay can be considered in the delay compensation independently from its position. This also leads to a higher absolute accuracy.

From a robotic point of view, SpaceWire is part of a heterogeneous network of different bus systems. Therefore it is also necessary to expand this general approach of globally synchronized local times to the other used communication systems which are responsible for data exchange with spatially separated units. Another issue from the control point of view is to compare the effects on noise and accuracy by comparing this approach with a system which spreads its values over the whole control cycle.

VI. ACKNOWLEDGMENT

This work was partly funded by Helmholtz Association within the Miro Innovation Lab project.

REFERENCES

- Stefan Jörg, Mathias Nickl, Alexander Nothhelfer, Thomas Bahls, and Gerd Hirzinger. The computing and communication architecture of the dlr hand arm system. In 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 1055–1062. IEEE, 2011.
- [2] Hermann Kopetz. Real-time systems: design principles for distributed embedded applications. Kluwer Academic Publishers, 1997.
- [3] Secretariat ECSS. Spacewire-links, nodes, routers and networks. Technical report, ECSS-E-ST-50-12C, Noordwijk, The Netherlands, July, 2008.
- [4] EtherCAT Technology Group. Ethercat protocol ehancements. Technical report, ETG.1020 S (R) V1.2.0, 2015.
- [5] FlexRay Consortium. Flexray communications system protocol specification version 3.0.1. Technical report, FlexRay Consortium, October, 2010.
- [6] Hermann Kopetz, Astrit Ademaj, Petr Grillinger, and Klaus Steinhammer. The time-triggered ethernet (tte) design. In Object-Oriented Real-Time Distributed Computing, 2005. ISORC 2005. Eighth IEEE International Symposium on, pages 22–33. IEEE, 2005.
- [7] Thomas Führer, Bernd Müller, Werner Dieterle, Florian Hartwich, Robert Hugel, and Michael Walther. Time triggered communication on can (time triggered can-ttcan). In 7th international CAN Conference, 2000.
- [8] U. Hagn, M. Nickl, S. Jörg, G. Passig, T. Bahls, A. Nothhelfer, F. Hacker, L. Le-Tien, A. Albu-Schäffer, R. Konietschke, M. Grebenstein, R. Warpup, R. Haslinger, M. Frommberger, and G. Hirzinger. The dlr miro: A versatile lightweight robot for surgical applications. *Industrial Robot: An International Journal*, 35(4):324–336, August 2008.
- [9] Mathias Nickl, Stefan Jörg, and Thomas Bahls. Towards high-speed spacewire links. In *Proceedings of the 5th International SpaceWire Conference*, pages 263–266, 2013.
- [10] Mathias Nickl, Stefan Jörg, Thomas Bahls, Alexander Nothhelfer, and Stefan Strasser. Spacewire, a backbone for humanoid robotic systems. In *Proceedings of the 4th International SpaceWire Conference*, pages 356–359. Space Technology Centre, University of Dundee, 2011.
- [11] R. Ginosar. Fourteen ways to fool your synchronizer. In Ninth International Symposium on Asynchronous Circuits and Systems, 2003. Proceedings., pages 89–96, May 2003.

Efficient Implementation of On-Chip Communication Optimized for SpaceWire Networks

Networks and Protocols, Short Paper

Markus Bihler and Thomas Bahls Institute of Robotics and Mechatronics German Aerospace Center (DLR) Münchenerstr. 20 82234 Weßling, Germany Markus.Bihler@dlr.de Thomas.Bahls@dlr.de

Abstract—SpaceWire has been used successfully as communication backbone implemented on Field Programmable Gate Arrays (FPGAs) in several robotic systems at the DLR. However, the growing number of available System-on-Chip (SoC) solutions with integrated programmable logic and the demand for efficient processing power in embedded systems require high speed On-Chip communication.

Using SpaceWire as interconnection bus inside FPGAs has a major drawback compared to dedicated On-Chip bus systems. As it is not designed to exploit the wide parallelism of FPGAs, several On-Chip communication standards outperform SpaceWire in terms of bandwidth and resource costs. Examples are the AMBA AXI bus, the Avalon bus, CoreConnect or the Wishbone SoC Interconnection. In this work, the Wishbone standard is used to show, how an FPGA system connected to a SpaceWire network can benefit from a dedicated On-Chip bus.

The architecture of the Wishbone standard allows an almost seamless integration into a SpaceWire network. Essential parts of SpaceWire, like package oriented communication or timecode distribution, can be mapped onto the Wishbone standard. A slim protocol is presented to allow accessing the On-Chip address space from the SpaceWire network and vice versa. It is also shown, that a heterogeneous On-Chip network consisting of SpaceWire and Wishbone occupies less resources on an FPGA than a similar network implemented with SpaceWire only. In the future, this approach can be used to integrate SoCs with built-in programmable logic into a SpaceWire network.

I. INTRODUCTION

SpaceWire is a slim and powerful standard for local area networks and inter chip communication [1]. At the DLR, it has been used successfully as communication backbone implemented on Field Programmable Gate Arrays (FPGAs) in several robotic systems, for example the DLR Miro [2], the DLR Mica [3] or the DLR Hand Arm System [4].

As the most complex system the DLR Hand Arm System is an anthropomorphic impact tolerant robot based on variable stiffness actuators. Its communication infrastructure comprises 52 actuators and 430 sensors in one arm. The respective SpaceWire topology is shown in Figure 1. To guarantee main control loop frequencies of up to 10 kHz a deterministic behavior, low latency, high bandwidth and mechanisms to synchronize the actuators and sensors are indispensable. Therefore, proprietary physical and character layers as well as an additional transport layer have been used to adapt SpaceWire to these requirements, as described in [5]. The physical and character layers provide a 1 Gbit/s link. The transport layer introduces the Datagram protocol for simple non-reliable connections as well as the RequestResponse protocol. For cyclic data, such as the desired and actual values of the robot's control loop, the Datagram protocol is used. It consists of a single SpaceWire packet, which is validated by a cyclic redundancy check (CRC). For asynchronous data, such as house keeping or configuration, the RequestResponse protocol is used. It consists of a request packet and a response packet, both validated by a CRC.



Fig. 1. SpaceWire topology of the DLR Hand Arm System; Octagons denote routing switches, squares denote nodes and rectangles denote physical links between routing switches

The DLR Hand Arm System's complex communication infrastructure requires a transport layer implementation in each SpaceWire node to be able to identify the source of each packet and to configure different communication targets for the SpaceWire nodes. The communication of up to 11 actuators and 15 sensors is handled by one FPGA. Therefore, using one SpaceWire node for each sensor or actuator results in a hardware design which does not fit on the used Xilinx Virtex 5 XC5VLX50 FPGAs. Too much resources are used by the SpaceWire router and the SpaceWire nodes inside the FPGA. To reduce the required resources, the communication of several sensors and actuators have been merged into fewer modules. These modules are not as easy to reuse because they are tailored to a specific set of sensors and actuators. Additionally, some flexibility in the communication is lost due to the fact that all data of one packet from such a module has to be sent to the same target in the SpaceWire network.

In general, hardware development on FPGAs is very time consuming compared to software programming. Not used in the DLR Hand Arm System but highly desirable for future robots and other systems are System-on-Chip (SoC) solutions with integrated programmable logic and one or more micro controllers. The combination of high level programming and hardware acceleration in SoCs can extend the system's flexibility and decrease the hardware development effort at the same time by reusing the same hardware accelerators for different applications. For a robotic system, this is an important step away from power demanding host computers towards efficient on board computing and more complex autonomous systems. However, such an implementation requires high speed On-Chip communication, that SpaceWire is not designed to provide.

Hence, using SpaceWire as interconnection bus inside FPGAs has a major drawback compared to dedicated On-Chip bus systems. It is not designed to exploit the wide parallelism of FPGAs. Therefore, several On-Chip communication standards outperform SpaceWire in terms of bandwidth and resource costs. Examples are the AMBA AXI or AHB buses¹, the Avalon bus [6] or the WISHBONE SoC Interconnection (Wishbone) [7]. Those On-Chip bus systems are specifically tailored for FPGA logic due to their distributed multiplexer approach. They reach higher data rates compared to SpaceWire while reducing the resources required by the communication network on the FPGA.

In this work, the Wishbone standard is used as On-Chip bus inside one of the DLR Hand Arm System's FPGAs, connecting data collecting modules with each other and the robot's SpaceWire communication backbone. One reason for the use of the Wishbone standard in this work is it's architecture, which allows an almost seamless integration into the SpaceWire network. Additionally, the Wishbone standard has a slim feature list dedicated to FPGA and ASIC implementations. Compared to AMBA AXI or AMBA AHB, this allows simpler implementations which are smaller in terms of hardware utilization. Another benefit of the Wishbone standard is it's public domain license. It is free to use and vendor independent.

This work presents a concept in which time-codes and

the packet oriented communication of SpaceWire have been mapped onto the Wishbone standard. By means of two example implementations it is shown that the use of Wishbone as On-Chip bus in a SpaceWire network has both, a smaller hardware footprint and a higher data rate, compared to using SpaceWire as On-Chip bus. In this work, 20% of an FPGAs resources have been saved and a maximum possible data bandwidth of 2.9 Gbit/s has been reached inside the FPGA.

In Section II, brief overviews of the Wishbone standard and the overall communication concept of SpaceWire robots at DLR are given. Afterwards, Section III describes how the Wishbone standard has been integrated into a SpaceWire network. The experimental results are presented in Section IV, which describes two example implementations for one of the DLR Hand Arm System's FPGAs, one with SpaceWire as On-Chip communication system and the other with a heterogeneous implementation consisting of SpaceWire and Wishbone. A conclusion is given in Section V.

II. BASICS

a) Wishbone: The Wishbone standard provides a common interface description for IP cores, optimized for FPGA and ASIC applications. It follows a master and slave architecture and allows a variety of network topologies, such as a point-to-point connection between one master and one slave, shared bus or cross bar switch interconnections. As the implementation of an interconnection module (*Intercon*) is not defined in the Wishbone standard, application specific solutions are possible.

 TABLE I

 Used signals of the Wishbone interface; rows highlighted in

 dark gray denote obligatory signals; rows highlighted in

 light gray denote user defined signals

Master	Slave	Name	Description
CYC_O	CYC_I	Cycle	Indicates a valid bus cycle
	TCD_I	Time-Code	Cycle tag signal
STB_O	STB_I	Strobe	Qualifies master signals e.g. ADR_O and DAT_O
WE_O	WE_I	Write Enable	Indicates write access
ADR_O	ADR_I	Address	8,16,32 or 64 bit bus
DAT_O	DAT_I	Write Data	8,16,32 or 64 bit bus
EOP_O	EOP_I	EOP	Master's end of packet marker (data tag type)
EEP_O	EEP_I	EEP	Master's error end of packet marker (data tag type)
ACK_I	ACK_O	Acknowledge	Qualifies slave signal DAT_O; ends a bus cycle
DAT_I	DAT_O	Read Data	8,16,32 or 64 bit bus
RTY_I	RTY_O	Retry	Marks slave as busy; Ends a bus cycle
EOP_I	EOP_O	EOP	Slave's end of packet marker (data tag type)
EEP_I	EEP_O	EEP	Slave's error end of packet marker (data tag type)

Important features of the Wishbone standard for this work are optional and user defined signals. Besides reset and clock signals, the Wishbone interface consist of 3 obligatory signals (see dark gray rows in Table I). With theses signals, the basic Wishbone handshaking protocol can be implemented.

¹Specifications for the AMBA protocols can be obtained from developer.arm.com



Fig. 2. Block diagram of a Communication Bridge; Yellow blocks denote SpaceWire and blue blocks denote Wishbone modules; Arrows indicate the direction of the data flow

The optional signals defined by the Wishbone standard can be used to tailor the implementation of the Wishbone bus to the application's requirements. Additionally, it is allowed to introduce user defined signals, which must be assigned to one of the three different tag types: the address tag, the data tag or the cycle tag (see rows highlighted in light gray in Table I). In general, optional output signals of the master are qualified by the master's *Cycle* and *Strobe* signals while optional output signals of the slave are qualified by the slave's *Acknowledge* signal. One important exception are cycle tag user signals, which are qualified only by the master's *Cycle* signal. In this work, this feature is used to distribute SpaceWire time-codes via Wishbone slave interfaces. In Table I, all interface signals used in this work are listed.

b) Robotic communication concept: In general, robots using SpaceWire at DLR follow a common concept for control data as described in [5]. One or more real-time host computers run a control model for the whole robotic system or separate parts of it. The input for such control models is called actual data (e.g. sensor values). The output of a control model is called desired data (e.g. commands for a motor controller). One control loop iteration consists of the input of actual data and the calculation and output of desired data. The iteration of one control loop is initiated by a time-code that is generated periodically.

III. CONCEPT

This work's concept enables data packets as well as timecodes to be exchanged via *Communication Bridges* between SpaceWire and Wishbone networks, as shown in Figure 2. In the Wishbone network, the time-codes can be distributed from Wishbone Masters via the *Intercon* to Wishbone slaves, e.g. in order to simultaneously start reading sensor data. Additionally, the time-codes can be used to manipulate the *Intercon's* arbitration behavior. Overall, this concept is tailored to DLR's robots using SpaceWire. Many design decisions have been made according to these robots' requirements and architecture. For example, data are pushed from the SpaceWire to the Wishbone network and vice versa. More complex transactions like read requests or a read-modify-write access can be implemented on higher communication layers. This allows a more simple implementation in the hardware. The following paragraphs describe the concept in detail.

In this work the Datagram protocol presented in [5] is used to send data to and from the *Communication Bridge*. However, a small extension has to be applied to the Datagram protocol by inserting the Wishbone address before the payload data as shown in Figure 3. Here, the Wishbone address represents the address of the first data word in the payload data. Another possibility to connect a *Communication Bridge* to a SpaceWire network is the Remote Memory Access Protocol (RMAP). However, for compatibility reasons and to reduce the implementation effort for DLR's robotic systems with SpaceWire, it is not used.

		1, 2, 4 or 8 bytes			-
SpW address	Datagram ID	Wishbone address	Payload data	CRC	EOP

Fig. 3. Structure of a Datagram packet containing Wishbone data

The *Communication Bridge* has two dedicated data paths to transfer data and time-codes from a SpaceWire to a Wishbone network and vice versa. For the first data path, the *Sink Node*, the *Write Master* and the *Time-Code Master* are used (see Figure 2). The *Sink Node* performs a CRC check on received Datagram packets and forwards them to the *Write Master* via a FIFO. Time-codes received from the SpaceWire network are sent to the *Time-Code Master*. Dual-clock FIFOs are used to decouple the two networks and allow different clock domains. To transfer data and time-codes from a Wishbone to a SpaceWire network, the *Read Slave* and the *Source Node* are used. The *Read Slave* receives Wishbone data, addresses and time-codes and forwards them to the *Source Node*, which creates a Datagram packet and calculates the CRC value.

To distribute SpaceWire time-codes from an *Intercon* to all its connected slaves simultaneously, a dedicated *Time-Code Slave* is used, which controls the *Intercon's Cycle* and *Time-Code* signals. When the *Time-Code Slave* has received a timecode, it raises the *Cycle* signals of all slaves connected to the *Intercon*. During that period, all user defined cycle tag signals including the *Time-Code* signal are valid. In this way, the *Time-Code* signal can be used to transfer time-codes to all connected slaves simultaneously without affecting or being delayed by ongoing Wishbone transactions.

To emulate the priority of time-codes in SpaceWire networks, sending a time-code to the *Time-Code Slave* may not be delayed by ongoing transactions. To achieve this without introducing new protocols, a dedicated Wishbone network can be used for the *Intercon Slave*. In Figure 2, this is implemented by a point-to-point connection between the *Time-Code Master* and the *Intercon Slave*. If there are more than one possible time-code sources, an additional *Intercon* can be used.

Data transactions to a Wishbone master can only be initiated by the master itself. Therefore, a module which is connected to the *Intercon* only via a master can not receive time-codes with the presented concept. An additional Slave is required.



Fig. 4. Block diagram of FPGA design with SpaceWire (implementation *a*)); Arrows indicate the direction of the data flow

To provide the means to control a master, time-codes are also used to control the *Intercon's* arbitration process. For example, a master could start Wishbone transactions continuously. The *Intercon* only connects the master to the addressed slave once, after a time-code has been received. Which master's communication is bound to time-codes can be configured with a look-up table. Via the *Config Slave*, the look-up table can be configured.

IV. RESULTS

To compare this work's concept with a SpaceWire network implementation without dedicated On-Chip bus, two similar firmware implementations for the same electronics are presented. The used digital electronics consist of a Xilinx Virtex 5 XC5VLX50 FPGA chip which is connected to two physical Gbps SpaceWire ports as described in [5] and thirteen cable connectors, e.g. used for I2C, SPI, BiSS or other serial communication protocols. The purpose of these electronics is to control the communication between a host computer and 11 actuators and 15 sensors in the robots lower arm.

Figure 4 and 5 show block diagrams of two example implementations. Both use thirteen SPI-Masters, capable of receiving actual data and transmitting desired data. Although several other peripheral communication standards are used by the robot, for a better comparability only SPI is implemented in these examples. Figure 4 shows implementation a), in which each SPI-Master is directly connected to the SpaceWire Routing Switch via two SpaceWire nodes. Figure 5 shows implementation b, in which a Wishbone On-Chip bus with a *Communication Bridge* is used to connect the SPI-Masters to the SpaceWire Routing Switch. Both implementations use shared bus topologies for the SpaceWire Routing Switches as well as the Wishbone *Intercon*. In this way, implementation a) fits into the given FPGA and is still comparable to implementation b). Additionally, the same data flow is used in



Fig. 5. Block diagram of FPGA design with SpaceWire and Wishbone combination (implementation b)); Yellow blocks denote SpaceWire and blue blocks denote Wishbone modules; Arrows indicate the direction of the data flow

both example implementations. After receiving a time-code, the SPI-Masters transmit the currently buffered actual data to the SpaceWire network and the buffered desired data is transmitted to the peripherals via SPI.

The overall device utilization of the two example implementations show, that implementation b) with the presented concept of using Wishbone as On-Chip bus in a SpaceWire network requires two more BlockRAM modules but can save 24.5% of Configurable Logic Block (CLB) Slices on a Virtex 5 FPGA (see Table II). These released hardware resources can either be used e.g. for preprocessing tasks or to enable the use of smaller an cheaper FPGA chips in future electronics. Additionally, implementation b) reaches a higher possible clock frequency of 91 MHz compared to implementation a) with 76.4 MHz. This results in an On-Chip communication bandwidth of up to 2.9 Gbit/s when using a data bus width of 32 bit.

TABLE II Overall device utilization on a Xilinx Virtex 5 XC5VLX50; Source: Xilinx ISE 13.2 Mapping report

		Implementation a)		Implementation b)	
Resource	Available	Used	Util.	Used	Util.
CLB Slices	7200	6160	85.6%	4400	61.1 %
Slice Registers	28800	16576	57.6%	11260	39.1 %
Slice LUTs	28800	18175	63.1 %	12910	44.8 %
Block RAMs	48	4	8.3 %	6	12.5 %

A more detailed utilization of single modules for both example implementations is given in Table III. It shows that the SPI-Masters with Wishbone masters and slaves in implementation b) are smaller by 40 % compared to the SPI-Masters with SpaceWire nodes in implementation a). Additionally, the Wishbone *Intercon* is smaller than the SpaceWire Routing Switch in implementation a) by a factor of 4, although it has a comparable number of ports. The *Intercon* is mostly implemented with LUTs, which represent its multiplexing structure on the FPGA.

TABLE III MODULE LEVEL UTILIZATION AFTER SYNTHESIS; SOURCE: MENTOR GRAPHICS PRECISION 2010A2 AREA REPORT

Module name	Slice Registers	Slice LUTs	Block RAMs
Implementation <i>a</i>)			
SPI-Master (13x)	876	969	0
SpaceWire Routing Switch	2797	3346	0
Implementation b)			
SPI-Master (13x)	523	567	0
Wishbone Intercon	4	789	0
Communication Bridge	289	342	2
SpaceWire Routing Switch	1244	1541	0

V. CONCLUSION

A concept to integrate an On-Chip communication bus into a SpaceWire network was presented. In this work the Wishbone standard was used as it allows an almost seamless integration into a SpaceWire network. It has been shown that SpaceWire's packet oriented communication as well as time-codes can be mapped into a Wishbone network. Time-codes can be distributed simultaneously to Wishbone Slaves connected to the same *Intercon*. Additionally, time-codes can be used to control the *Intercon's* arbitration process which allows to control the communication flow of Wishbone Masters.

The presented concept allows more resource efficient implementations on FPGAs compared to solutions using SpaceWire as On-Chip bus. This was confirmed by comparing two example implementations for the same digital electronics. Both examples implement the same functionality while one uses the presented concept and the other is based on SpaceWire. It was shown that the use of Wishbone as On-Chip bus can save more than 20% of an FPGA's resources and provide higher data rates compared to a design that uses SpaceWire as On-Chip communication bus.

Additionally to those quantifiable benefits, the use of an On-Chip bus such as Wishbone in a SpaceWire network provides a common interface that can be used throughout a robotic system. On the one hand, this eases the collaboration of large developer teams which work on the same project. On the other hand, a common interface enhances the reusability of designs and allows an easier integration of third party IP cores.

In future work the presented concept could for example be used to integrate implementations of control models for parts of a robotic system in SoC FPGAs. This would relieve both, the host computers and the communication network between robotic systems and hosts. As a result, on the one hand the robotic system becomes more independent of its host computer. On the other hand, higher control loop frequencies become possible.

REFERENCES

- European Cooperation for Space Standardization (ECSS). (2008) SpaceWire - Links, nodes, routers and networks. [Online]. Available: http://ecss.nl/standard/ecss-e-st-50-12c-spacewire-links-nodesrouters-and-networks/
- [2] U. Hagn, M. Nickl, S. Jörg, G. Passig, T. Bahls, A. Nothhelfer, F. Hacker, L. Le-Tien, A. Albu-Schäffer, R. Konietschke, M. Grebenstein, R. Warpup, R. Haslinger, M. Frommberger, and G. Hirzinger, "The DLR MIRO: A versatile lightweight robot for surgical applications," *Industrial Robot*, vol. 35 (4), pp. 324 –336, 2008. [Online]. Available: http://elib.dlr.de/76858/
- [3] S. Thielmann, U. Seibold, R. Haslinger, G. Passig, T. Bahls, S. Jörg, M. Nickl, A. Nothhelfer, U. Hagn, and G. Hirzinger, "MICA - A new generation of versatile instruments in robotic surgery," in *IROS* 2010, *IEEE International Conference on Intelligent Robots and Systems*, October 2010. [Online]. Available: http://elib.dlr.de/68322/
- [4] M. Grebenstein, A. Albu-Schäffer, T. Bahls, M. Chalon, O. Eiberger, W. Friedl, R. Gruber, S. Haddadin, U. Hagn, R. Haslinger, H. Höppner, S. Jörg, M. Nickl, A. Nothhelfer, F. Petit, J. Reill, N. Seitz, T. Wimböck, S. Wolf, T. Wüsthoff, and G. Hirzinger, "The DLR Hand Arm System," in *ICRA 2011*, May 2011. [Online]. Available: http://elib.dlr.de/70101/
- [5] M. Nickl, S. Jörg, T. Bahls, A. Nothhelfer, and S. Strasser, "SpaceWire, A Backbone For Humanoid Robotic Systems," in *International SpaceWire Conference*. Space Technology Centre, University of Dundee, November 2011, pp. 356–359. [Online]. Available: http://elib.dlr.de/73207/
- [6] Intel. (2017) Avalon Interface Specifications MNL-AVABUSREF 2017.05.08. Intel. [Online]. Available: https://www.altera.com/en_US/pdfs/literature/manual/mnl_avalon_spec.pdf
- [7] OpenCores. (2010) Wishbone B4 WISHBONE System-On-Chip (SoC) Interconnection Architecture for Portable IP Cores. OpenCores. [Online]. Available: http://cdn.opencores.org/downloads/wbspec_b4.pdf

Feasibility Study of Wireless Communication System Operating on SpaceWire Network

Session: Networks and Protocols, Short Paper

Makoto Tomitaka, Yasushi Igarashi, Satoshi Ichikawa, Noriyasu Inaba Research and Development Directorate Japan Aerospace Exploration Agency (JAXA) 2-1-1, Sengen, Tsukuba, Ibaraki 305-8505, Japan tomitaka.makoto@jaxa.jp, igarashi.yasushi@jaxa.jp, ichikawa.satoshi@jaxa.jp, inaba.noriyasu@jaxa.jp

Atsushi Tomiki, Keiichi Matsuzaki Institute of Space and Astronautical Science (ISAS) Japan Aerospace Exploration Agency (JAXA) 3-1-1, Yoshinodai, Sagamihara, Kanagawa 252-5210, Japan tomiki.atsushi@jaxa.jp, matsuzaki.keiichi@jaxa.jp

Abstract — We demonstrate a wireless communication system operating on a SpaceWire network for the first time. It is difficult to directly replace the SpaceWire network with a wireless communication system, as each concept of network architecture is different except for point-to-point connection. However, we realized a wireless communication system on the SpaceWire network by controlling the functionality of Remote Memory Access Protocol (RMAP). Each node of this wireless communication system can communicate with other nodes via wireless connections as well as conventional wired connections such as using SpaceWire cables. We adopted impulse radio ultrawideband (IR-UWB) technology in a physical layer of this wireless communication system, which is robust against multipath propagation due to reflected waves inside a spacecraft as a metalenclosed environment. This paper briefly reports on a feasibility study conducted for the above hybrid SpaceWire system between wireless and wired communication.

Index Terms-Wireless, UWB, IR-UWB, SpaceWire, RMAP

I. INTRODUCTION

The Japan Aerospace Exploration Agency (JAXA) is addressing the challenges of innovating wireless technologies for spacecraft. For example, adopting wireless technologies within a spacecraft could contribute to: (i) a reduction in the mass, cost, and scheduling of spacecraft, (ii) design flexibility in the layout of spacecraft subsystems, and (iii) improvement of safety and workability in Assembly, Integration and Test (AIT). This feasibility study focused on a medium data-rate wireless communication capability (e.g. above 250 kb/s but less than 10 Mb/s) that operates in a metal-enclosed spacecraft.

To utilize wireless technology in spacecraft, many studies have reported the measurements and characterizations of radio propagation testing in a heavy multipath environment [1]-[4]. Ryouhei Kobayashi, Minoru Kumakiri, Iwao Fujishiro Shimafuji Electric Inc. 6-36-11, Nishikamata, Ota-ku, Tokyo 144-0051, Japan

kobayashi@shimafuji.co.jp, kumakiri@shimafuji.co.jp, fujishiro@shimafuji.co.jp

Masaharu Nomachi

Research Center for Nuclear Physics, Osaka University 1-1, Machikaneyamacho, Toyonaka, Osaka 560-0043, Japan <u>nomachi@rcnp.osaka-u.ac.jp</u>

These reports suggest that ultra-wideband (UWB) systems have an advantage over narrowband in terms of reducing the fading margins for robust and noninvasive short-range communications. The legacy UWB system entails carrierless wireless communication and is completely different from current narrowband or spread-spectrum wireless communication architecture. The approved frequency according radio regulations for commercial use ranges from 3.1 GHz to 10.6 GHz at a bandwidth of 7,500 MHz. As the emission power in that frequency range is limited to a maximum of -41.3 dBm/MHz, we expect that the UWB system will contribute to reducing the potential EMC issues within spacecraft. There are two types of UWB technology: impulse radio UWB (IR-UWB) and orthogonal frequency division multiplexing UWB (MB-OFDM UWB). Although many leading companies have recently declined to promote the MB-OFDM UWB system, the IR-UWB system has emerged as a promising technology for short-range data communication and indoor localization [5]-[7]. We have thus decided to adopt the IR-UWB system for the purpose of wireless communication within the spacecraft.

As a first step, we adopt SpaceWire protocol in an upper layer of the wireless communication system to extract the requirements for its application to spacecraft. This paper reports the results of a feasibility study conducted for a wireless communication system that operates on a SpaceWire network and which is to be installed in a metal-enclosed spacecraft.

II. IR-UWB SYSTEM

A. IR-UWB System Specifications

JAXA and GIT Japan Inc. have jointly developed an IR-UWB system that can operate within a metal-enclosed spacecraft. Table I lists the main specifications of the IR-UWB system. In this study, the IR-UWB system operates at a highband frequency (7.25 to 10.25 GHz) approved by Japanese radio regulation. One important parameter when implementing IR-UWB is the pulse repetition frequency (PRF), which denotes the number of pulses occurring per second.

A commercial IR-UWB system cannot ensure to communicate within a heavy multipath environment such as a shield box. Our IR-UWB system implemented a function to reduce the influence of reflected waves. Moreover, our IR-UWB system can operate with two modulation schemes: Pseudo Random Noise (PN) and PN-Pulse Density Modulation (PN-PDM). The PN-PDM scheme modulates 1 bit into 4 bits with a certain code after PN modulation, and is an invention of GIT Japan Inc.

Item	Specification
Frequency Band	7.25 to 10.25 GHz (High band approved in Japan)
Transmit Output Power	-41.3 dBm/MHz (max.)
Antenna	Bi-Conical Antenna: 0 dBi
Pulse Repetition Frequency (PRF)	50 MHz / 25 MHz / 12.5 MHz
Modulation Scheme	Pseudo Random Noise (PN) / PN- Pulse Density Modulation (PN-PDM)
PN Code	M-sequence 8 / 16 / 32
Maximum Transmission Unit (MTU)	256 / 512 / 1024 / 2048
Error Collection	ECC (223, 255)

The parameter settings indicated in **bold type** are used in testing.

B. IR-UWB System Performance

We tested the performance of our IR-UWB system within a heavy multipath environment by using a shield box with dimensions of 480 mm (W) \times 480 mm (D) \times 480 mm (H). Figure 1 shows the test setup. The IR-UWB system subjected to testing consists of two IR-UWB modules. Each of the two antennas mounted in the shield box is connected to one IR-UWB module placed outside the shield box via a feedthrough panel. The top lid was closed during the test. Table 1 shows the parameter settings (in bold type) used for this test.

We evaluated the Bit Error Rate (BER) and throughput of the IR-UWB system operating with the PN and PN-PDM modulation schemes. The BER does not include corrected errors. Figure 2 shows the results of the performance test.

All BERs for the PN modulation scheme are higher than 10^{-6} (Fig. 2 (a)) and many errors are detected. The throughput rate measured at 50 MHz PRF is lower than the estimated throughput rate (5.14 Mbps < 5.28 Mbps), but there is little degradation of throughput rate at 12.5 MHz PRF or 25 MHz PRF. It is therefore necessary to make the IR-UWB system more robust against multipath propagation when using the PN modulation scheme.

In contrast, all BERs for the PN-PDM modulation scheme are less than 10⁻⁹ and a few errors are detected. The throughput rate measured at 50 MHz PRF is approximately 1.33 Mbps. This

result shows that the IR-UWB system operating with the PN-PDM modulation scheme can communicate within a heavy multipath environment by adopting the retry function.



Fig. 1. IR-UWB performance test setup



Fig. 2. Performance of the IR-UWB system in a heavy multipath environment

III. HYBRID SPACEWIRE SYSTEM DESIGN

A. Hardware Design

Figure 3 shows a block diagram of the Wireless Communication Unit (WCU), which consists of an IR-UWB board, a SpaceWire (SpW) board, and a data acquisition (DAQ) board. The SpaceWire IP core designed by Shimafuji Electric Inc. is embedded into an FPGA on the SpW board. To realize a hybrid SpaceWire system, the SpW board uses the four-port SpaceWire Router: Port-1 and Port-2 are for external SpaceWire-enabled equipment, Port-3 is for the DAQ board, and Port-4 is for the IR-UWB board. The DAQ board used for demonstration of this system has the following functions: 320 x 240 (QVGA) video streaming and ethernet connection.



Fig. 3. Block diagram of the Wireless Communication Unit (WCU)

B. Network and Protocol Design

It is impossible to directly replace a SpaceWire network with a wireless communication system for the following reasons:

- The architectural concept for SpaceWire entails a pointto-point topology, while the architectural concept for wireless communication entails a peer-to-peer topology.
- 2. SpaceWire-enabled equipment operates at full duplex, while wireless equipment operates at half duplex.

Accordingly, we modified the SpaceWire protocol [8], [9]. Figure 4 shows the protocol stack of wireless communication over the SpaceWire network. The details are given below.

1) Network Layer

The network layer in the SpW board supports the receiving/sending of RMAP packets via the DAQ board or external equipment in the same way as SpaceWire protocol. When the destination address of an RMAP packet is the equipment on another SpaceWire network or a wireless node, the network layer generates an IR-UWB packet that adds the UWB header and SpaceWire packet length to the RMAP packet. When the destination address received from the IR-UWB board is the equipment on its own SpaceWire network, the network layer formats the RMAP packet according to the UWB header and SpaceWire packet length. As described above, we realize a peer-to-peer topology over the SpaceWire network.

2) Data Link Layer

The data link layer in the SpW board supports TX and RX control as under SpaceWire protocol, except for the quality of service (QoS) as TX/RX priority because the wireless equipment operates at half duplex. Basically, it is possible to operate using existing SpaceWire protocol because the IR-UWB

system is configured for Time Division Multiple Access (TDMA) systems. However, the TX/RX priority function is necessary in case of such unexpected events as multiple access. In this study, the TX data was set to higher priority. However, we will modify the QoS function including a retry function in the future.

3) Encoding Layer

The encoding layer embedded in the IR-UWB board adds IR-UWB specific functions, such as modulation/demodulation, CRC, and ECC.



Fig. 4. Protocol stack of wireless communication over the SpW network

C. Proposed Hybrid SpaceWire System

Figure 5 shows the proposed hybrid SpaceWire system architecture. By implementing the above-mentioned designs, communication with multiple wireless nodes becomes possible by connecting only one node to the SpaceWire network with a conventional SpaceWire cable. In addition, wireless equipment behaves like a gateway in a wireless communication system that connects one SpaceWire network with another SpaceWire network.



Fig. 5. Proposed hybrid SpaceWire system architecture

IV. EVALUATION OF THE HYBRID SPACEWIRE SYSTEM

A. System Validation

We first validated the hybrid SpaceWire system while not in a multipath environment. Figure 6 shows the configuration of the validation test. The IR-UWB system operated with PN-PDM modulation at 50 MHz PRF.



Fig. 6. Test configuration of system validation

1) Functional Test

This test is for validating the network and link layer design. The camera data acquired by WCU2 is transferred to WCU1 and WCU3. Table II shows the settings and results. In all test cases, all WCUs properly identified the destination logical address and operated as expected.

TABLE II. FUNCTIONAL TEST CASES AND RESULTS

	Settings				Results	
Unit Name (Mode)	WCU2 (TX)	WCU1 (RX)	WCU3 (RX)	WCU1 (RX)	WCU3 (RX)	Judge
Logical	FE	FE	FD	R	N	Passed
Address	FE	FD	FE	Ν	R	Passed
	FE	FE	FE	R	R	Passed

FE and FD denote the logical address. R: Received, N: Non-received

2) Loopback Test

This test is for validating the quality of data during longterm operation. WCU1 transfers the incremental data of 655,360 bytes (i.e. the SpaceWire cargo) to WCU3 using a SpaceWireto-GigabitEther Unit (SGU) shown in Fig. 6. The PC stores the data received from WCU3 via the SGU. The PC also monitors the throughput rate and BER of the hybrid SpaceWire system. During the test, the throughput rate is approximately 1.30 Mbps and no bit error occurs. Figure 7 shows that no error is detected as a result of comparing the transmitted data and received data.



Fig. 7. Comparison result between TX data and RX data in loopback test

B. Measurement Results in Heavy Multipath Environment

1) Test Setup

As shown in Fig. 8, the performance test was conducted using three WCUs and a structural thermal model (STM) of a small scientific satellite that was launched in 2009. The STM is a rectangular parallelepiped, measuring 700 mm (W) \times 700 mm (D) \times 600 mm (H) in size. The inside of the STM is divided into two rooms across the partition. The test configuration is the same as shown in Fig. 6 except for the antenna locations. The antennas were located in either line of sight (LOS) or non-LOS (NLOS) of the STM: Two antennas of WUC1 and WCU2 on the one side (LOS: Fig. 8 (b)) and one antenna of WCU3 on the other side of the STM (NLOS: Fig. 8 (c)) were located in the STM. Each antenna was connected to each WCU placed outside the STM by using cables.



(b) Antenna locations on LOS side (c) Antenna location on NLOS side

Fig. 8. Evaluation test setup using STM

2) Measurement Results

The IR-UWB system operated with the PN-PDM modulation scheme at 50 MHz PRF during the test. Table III shows the settings and results. When WCU1 or WCU3 is TX, the incremental data is transmitted using the SGU. When WCU2 is TX, the data taken by the camera itself is transmitted.

In case of LOS, all BERs are less than 10^{-8} , and thus the throughput rates become the estimated value (1.30 Mbps). In case of NLOS, many errors are detected so that the throughput

rates are less than the estimated value. These results suggest that it is possible to adopt this hybrid SpaceWire system within a metal-enclosed spacecraft by locating the antennas in LOS condition.

TABLE III.	PERFORMANCE	TEST CASES	AND RESULTS
------------	-------------	------------	-------------

Tost	Settings			Results	
Case	TX	RX	Propagation Path	Throughput Rate	BER
1-1	WCU1	WCU2	LOS	1.30 Mbps	3.53E-09
1-2	wcui	WCU3	NLOS	1.24 Mbps	7.92E-05
2-1	WGUD	WCU1	LOS	1.30 Mbps	0.00E+00
2-2	WCU2	WCU3	NLOS	1.28 Mbps	4.74E-05
3-1	WOU2	WCU1	NLOS	0.93 Mbps	2.45E-04
3-2	webs	WCU1	NLOS	1.01 Mbps	1.29E-06

V. CONCLUSION AND FUTURE WORK

We demonstrated a wireless communication system over a SpaceWire network for the first time. We adopted the IR-UWB system in the physical layer of the wireless communication system. We also modified SpaceWire protocol to realize this system. This phase of study suggested that it is possible to adopt this hybrid SpaceWire system within a metal-enclosed spacecraft when the antennas of the IR-UWB system are located in line of sight (LOS) within such spacecraft.

In order to improve the BER degradation even if antennas are located in non-LOS (NLOS), we are addressing ways to make the IR-UWB system more robust against multipath propagation. Moreover, we plan to incorporate the following protocols into the hybrid SpaceWire system: (i) robust functions (i.e. acknowledgement and retry) and (ii) time division multiple access (TDMA) communication, time synchronization, and broadcast.

ACKNOWLEDGMENT

This work was conducted in conjunction with GIT Japan Incorporated, which supported the development of the IR-UWB system as the physical layer of this wireless communication system. Furthermore, we would like to thank JAXA's wireless team and Japanese SpaceWire user members for their useful discussions and wide support.

REFERENCES

- [1] A. Matsubara, T. Ichikawa, A. Tomiki, T. Toda and T. Kobayashi, "Experimental evaluation of ultra wideband wireless transmission within a spacecraft for replacing wired interface buses," in Proc. CANEUS Fly-by-wireless Workshop 2010 (FBW10), Orono, ME, USA, Aug. 2010.
- [2] A. Matsubara, A. Tomiki, T. Toda and T. Kobayashi, "Measurements and characterization of ultra wideband propagation within spacecraft—proposal of wireless transmission for replacing wired interface buses," In: Advances in Spacecraft Technologies, IN-TECH,978-953-307-551-8, pp.61-74, Vienna, Austria, EU, Feb. 2011.

- [3] S. Hamada, A. Tomiki, T. Toda and T. Kobayashi, "Wireless connections within spacecraft to replace wired interface buses," in Proc. 2013 IEEE Aerospace Conf., Montana, USA, Mar. 2013.
- [4] T. Kobayashi and M. Hirose, "Wideband and ultra wideband radio propagation in heavy multipath environments," IEICE Trans. Fundamentals, Vol. E98-A, No. 2, Feb. 2015.
- [5] <u>http://www.ieee802.org/15/pub/TG4a.html</u>
- [6] B. Uguen, M. Laaraiedh, B. Denis, J. Keignart, J. Stephan and Y. Lostanlen, "Extraction and characterization of locationdependent UWB radio features with practical implications for indoor positioning," European Wireless 2012, Poznan, Poland, April 2012.
- [7] Huan-Bang Li, Ryu Miura, Toshinori Kagawa, Fumihide Kojima and Hisashi Nishikawa, "Improvement of receiver sensitivity for enhancing IR-UWB performance for indoor positioning," 2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC), pp.1-5, October 2017.
- [8] ECSS-E-ST-50-12C, Space Engineering SpaceWire- Links, nodes, routers and networks, July 2008.
- [9] ECSS-E-ST-50-52C, Space Engineering SpaceWire-Remote memory access protocol, February 2010.

SpaceWire network system for ERG mission network and the result of its performance

Networks and protocols, Short Paper

Sadatoshi Eguchi

Institute of Space and Astronautical Science (ISAS) Japan Aerospace Exploration Agency (JAXA) Sagamihara-City, Kanagawa-Pref., Japan eguchi@stp.isas.jaxa.jp

Abstract-We present the network system for ERG mission network, which is currently operated on its orbit. The network topology is ring network. The link speed is 20Mbps and the maximum throughput is 12Mbps on the design. This throughput is achieved by "real-time control" and "two data transfer methods" that is adopted as upper layer of SpaceWire. "Realtime control" separates 1 second into 16 windows. This window is called as mission time slot. Mission time slot consists of 4 periods; FDIR period, prepare period, transfer period and idle period. With "real-time control", ERG mission network restricts data transfer error to one slot and assigns maximum delay time of data transfer. One of "two data transfer methods" is "Relay method"; all nodes on the network share data via "Relay packet". Two is "Direct method"; a node transfers data to another node via "Direct packet". Relay method supports high reliability and Direct method supports high speed transfer.

Index Terms-SpaceWire, networks and protocols, realtime

I. INTRODUCTION

ERG satellite (Arase) is a scientific satellite that directly observes space storms occurring in the outer space near the Earth (geospace) in order to elucidate the mechanism of the development of the space storm. The ERG satellite was launched on December 20, 2016 and continues observations successfully now.

Communications between devices equipped with ERG satellite is done via SpaceWire network. This SpaceWire network is divided into two network systems. One is "bus network" that connects devices that are responsible for the common functions of satellites, such as communication with ground station, attitude control and so on, and the other is "mission network" that connects observation instruments specialized for mission of ERG satellite.

The mission network consists of nine observation instruments and a mission data processor (MDP) that manages them. The observation instruments and MDP perform as SpaceWire Node and also a maximum of 3 ports SpaceWire Router. The devices are connected via the adjacent SpaceWire Router, thereby configuring a mesh type SpaceWire network. Also, MDP performs as a bridge between these two networks.

This paper introduces the function of this mission network.

Takeshi Takashima, Iku Shinohara Institute of Space and Astronautical Science (ISAS) Japan Aerospace Exploration Agency (JAXA)

Sagamihara-City, Kanagawa-Pref., Japan ttakeshi@stp.isas.jaxa.jp, iku@stp.isas.jaxa.jp

II. ERG MISSION NETWORK PROTOCOL

The ERG mission network is based on the following design concepts.

- Real-time control; Make the network operate on a time unit called "Mission Time Slot" and close the communication processing in the slot. Even if an abnormality occurs in a certain "Mission Time Slot", it does not affect the next "Mission Time Slot".
- Two kinds of communication methods; One is Relay packet method; Reliable, slow to turn each device in turn and used for shared data between devices. The other is Direct packet method; Low reliability (single transmission, error handling with multiple packets), High-speed and mass data transfer possible.

The protocol stack of the mission network is shown in Fig.

Fig. 1. Protocol stack of the mission network.

A. SpaceWire Layer

1.

Function to exchange data between different devices. Compliant with ECSS SpaceWire standard[1].

However, in order to synchronize the spin of the satellite and the operation of the network nodes, the function is expanded for the SpaceWire Time-Code.

B. Real-time Control Layer

A function that guarantees the maximum delay time of packet transfer and limits the influence range of errors in the SpaceWire network.

The synchronization method of devices is also specified by this layer.

C. RMAP Layer

A function for allowing devices to read and write memory data via SpaceWire network. Compliant with ECSS RMAP standard[2].

D. Network Layer

A function for implementing packet control (regulation of relay packet method, direct packet method), route control, bandwidth control and to perform data transmission without delay in accordance with the requested data generation rate.

E. Service Layer

A function to establish a series of communication procedures necessary for realizing command distribution / telemetry collection for applications or observation devices.

III. SPACEWIRE LAYER

Basically, Compliant with SpaceWire standard [1]. But the contents customized for the mission network are shown below.

A. Link Speed

The communication link speed operates at the initial link 20 MHz for both TX and RX and at 20 MHz after linking.

B. Network Configuration

In the network configuration, a mesh type that can secure redundancy of communication paths and has a small number of cables is adopted. However, it can also be applied to other network configurations such as a star type and a ring type.

C. Time-Code

We use the two most significant bits (T_6, T_7) of the Time-Code to discriminate Time-Code type. Fig. 2. shows how to discriminate Time-Code type. If the discrimination bit is 0, the Time-Code is treated as "Slot Time-Code", If the discrimination bit is 1, the Time-Code is treated as "Index Time-Code".



T7: Do not care (no expansion)

Fig. 2. Extension of Time-Code

The "Slot Time-Code" is used for synchronization between the network components. The Slot Time-Code is periodically distributed to all the devices from the MDP. The Time-Code value is incremented by +1 for each delivery.

The "Index Time-Code" is used to notify the timing of the index pulse which is generated in synchronization with the satellite's spin. The Index Time-Code is delivered to all the devices every time an index pulse is generated.

The management of the Slot Time-Code value and the Index Time-Code value are independently with each other.

IV. REAL-TIME CONTROL LAYER

In order to guarantee the maximum delay time of data transmission and limit the influence range of communication error, real time control is introduced to this network. In this network, real-time control is defined as "completing communication within the special time window". The window is called the "Mission Time Slot".

A. Mission Time Slot

The specifications of the "Mission Time Slot" are shown below.

- Set the Slot Time-Code whose Time-Code value is a multiple of 2ⁿ (n = 0 to 6) as "Time-Code separating the Mission Time Slot".
- A section from a certain "Time-Code separating a Mission Time Slot" to a next "Time-Code separating a Mission Time Slot" is called a "Mission Time Slot".
- The number of "Mission Time Slot" per one second shall be 2^{m.}

In the ERG mission network, let n = 2 and m = 4. Fig. 3. shows the mission time slots. The Slot Time-Code interval is 15.625 ms. That is, the period of the mission time slot is 62.5 ms.



Fig. 3. Mission Time Slots

B. Mission Time Slot Processing

The inside of the Mission Time Slot is divided into 4 periods; the FDIR period, the preparation period, the communication period, and the idle period. The relationship between each period is shown in Fig. 4.



Fig. 4. Periods of Mission Time Slot

The detail of each period is shown in below:

• FDIR period: The FDIR period is a period starting at the beginning of the Mission Time Slot and is a period for recovering from the error state in which the transmitted packet in the previous Mission Time Slot cannot be completed and remains on the network. A packet whose transmission cannot be completed before the next Mission Time Slot despite the start of transmission is called a "Staying Error Packet". Staying Error Packets are generated by routing loops (a state in which a packet passes through only the router and does not reach the node), a loss of EEP, or the like.

- Preparation period: The preparation period is a period next to the FDIR period, which is a period for acquiring the packet reception result in the previous Mission Time Slot and preparing for packet transmission in the current Mission Time Slot. In the preparation period, the following processing is performed; (1) Collection of FDIR processing result.
 (2) Completion processing of received packet in previous Mission Time Slot. (3) Preparation of sending and receiving packets at the current Mission Time Slot.
- Communication period: The communication period is a period next to the preparation period and is a period during which each node can freely communicate.
- Idle period: The idle period is the last period within the Mission Time Slot and is a period for absorbing the variation in the Mission Time Slot due to the generation accuracy of the synchronization signal from other networks and the like. The idle period is a period secured by design, so no processing is requested.

We referred to the guidelines of SpaceWire Real-Time Property Guarantee Methods by JAXA and Nagoya University.

V. RMAP LAYER

Compliant with ECSS RMAP standard[2]. It shall be within 10 μ s from the EEP of RMAP command reception until the start of RMAP reply transmission.

Read-Modify-Write Command is not used on the network.

VI. NETWORK LAYER

It implements packet control, route control, bandwidth control, and specifies a method for smoothly performing data transmission according to the required data generation rate.

A. Packet control

In this network, the 2 kinds of packets are used depending on the application.

Relay packet: A Relay Packet is a packet that patrols • each node in 1 second cycle. The Relay Packet starts from MDP, patrol each node and finally returns to MDP. The Relay Packet is transmitted as an RMAP Write. The outline of Relay Packet is shown in Fig. 5. The following is the process of the node which has received the Relay Packet: (1) Writes the local node ID to the list in the Relay Packet, which is named "Relay Confirmation List". (2) Waits until the next Mission Time Slot starts. (3) Overwrites the part of the Relay Packet by new data if application requests. (4) Sends the Relay Packet to the node which is implied in the list in the Relay Packet, which is named "Relay Order List". (5) If the node fails to send the Relay Packet, wait until the next time slot and send the relay packet to the node following the "Relay Order Table". The node that failed to receive the Relay Packet does not receive at this cycle and can receive at the next cycle. (Fig. 6.)







Fig. 6. Relay Packet transaction (error case)

• Direct packet: A Direct packet is a packet to be directly transmitted from an arbitrary node to a arbitrary node. Multiple packets can be transmitted in one slot, and a large amount of packet transmission is also possible. The Direct Packet is transmitted as an RMAP Write or an RMAP Read. The outline of Direct Packet is shown in Fig. 7. The following is the process of the node which is requested to send the Direct Packet: (1) Set Direct Packets to send buffer. (2) Waits until the next Mission Time Slot starts. (3) Starts the transmission of all Direct Packets stored in the send buffer. (4) If the node fails to send Direct Packet, Notices the application which requests. (Do not resend Direct Packet)



Fig. 7. Direct Packet method

B. Route control

In this network, the path address is used for route control of packets. A route between each node is defined by a path address.

The path address can be changed at an arbitrary timing by the application. The new route is applied from the next Mission Time Slot. If a communication error occurs due to disconnection or the like, the network can be recovered from the error by setting a detour route.

C. Bandwidth control

In order to avoid the congestion due to the excess of the communication volume of the whole network, the number of packets transmitted per Mission Time Slot is limited for each node. Since the entire network is regarded as one link and the number of packets is allocated, it is unnecessary to recalculate the number of packets even if it is necessary to change the route due to a failure or the like.

Band calculation is as below:

- The basic parameters of the network are as follows: link speed, Mission Time slot period, and total of FDIR period + preparation period + idle period.
- Relay packet parameters: Assign the following parameters: Relay packet size, and Number of relay packet transmission (fixed once).
- Direct Packet parameters: Assign the following parameters: Direct packet maximum size, and number of direct packet transmission, which is allocated for each node according to the data generation rate.
- Calculate the amount of the basic parameters, Relay packet parameters, and direct packet parameters. And confirm that the result of the calculation falls within one Mission Time Slot period. If the result of the calculation exceeds one Mission Time Slot period, it is not possible to transfer all the data to be transferred within one Mission Time Slot, so it is necessary to adjust the link speed, reduce the data amount, etc.



Fig. 8. Calculation of the bandwidth

VII. SERVICE LAYER

In the ERG mission network system, following services are supported.

• Command delivery service: Service for sending command packets to nodes via Relay Packets.

Commands to applications on each equipment are delivered via this service.

- Time distribution service: Service to broadcast 32-bit time data to all nodes via Relay Packets.
- Memory load service: Service for sending data from direct MDP to nodes via Direct Packets. It is used for device power supply ON / OFF, device startup, program rewrite, etc.
- Essential HK collection service: Service for collecting housekeeping data from nodes to MDP via Direct Packets. MDP sends Essential HK to the ground station via bus network. Essential HK is transmitted to the satellite bus every second.
- HK collection service: Service for collecting HK data from nodes to MDP via Relay Packets. Larger size HK data than essential HK can be transmitted to the satellite bus but not guaranteed outputting every second.
- Telemetry collection service: Service to transmit observation data generated by direct observation equipment to MDP / MDR.
- Memory dump service: Service for sending RMAP Raw Data from direct CPU board to MDP / MDR. It is used for program verification etc.
- Data share service: Relay a service for applications on the CPU board to exchange data by mediating networks. It assumes use of event notification etc. for SWPIA processing.

The relay packet data flow for each communication service is shown in Fig. 9. The direct packet data flow is shown in Fig. 10.



Fig. 9. Network services via Relay Packet



Fig. 10. Network service via Direct Packet

VIII. PEFORMANCE ON THE ORBIT

TABLE I. summarizes the performance of the ERG Mission Network. TABLE II. shows error counts of the ERG Mission Network. Both tables are statistics from Mar 2017 to 28th Feb 2018. Here, Relay Packet error is defined as the case that one or more nodes do not receive Relay Packet in 1 second. Direct Packet error is defined as the case that any nodes do not receive any reply packet implied with success status despite that the node send RMAP Command.

Error recovery processes were automatically performed as designed when each error is detected, so that it is not necessary to run a special operation (network reset, reconfiguration, etc.) without error clear commands.

TABLE I. PERFORMANCE OF ERG MISSION NETWORK

Item	Packet Type			
	Relay Packet	Direct Packet	Total	
Total packet count [packets]	$3.0 * 10^{7}$	$8.8 * 10^8$	9.1 * 10 ⁸	
Transferred data size [Gbytes]	$2.3 * 10^2$	$1.0 * 10^3$	$1.2 * 10^3$	

Item	Packet Type			
	Relay Packet	Direct Packet	Total	
Average of data rate [kbps]	64	290	350	
Maximum data rate [kbps]	64	7700	7800	

TABLE II. ERROR COUNT OF ERG MISSION NETWORK

Error	Count
Link disconnection (instantaneous)	37
Staying Error Packet detection	1
Relay Packet error	30
Direct Packet error	27

REFERENCES

- ESA Requirements and Standards Division, "ECSS-E-ST-50-12C SpaceWire - Links, nodes, routers and networks," 31 July 2008.
- [2] ESA Requirements and Standards Division, "ECSS-E-ST-50-52C Space engineering - SpaceWire - Remote memory access protocol," 5 February 2010.

Simulator for HIgh-speed Networks (SHINe): an OMNeT++ simulator for SpaceFibre and SpaceWire Networks

Session: Networks and protocols, Short Paper

Alessandro Leoni, Luca Fanucci Department of Information Engineering University of Pisa Via Caruso 16, 56122, Pisa, Italy alessandro.leoni@ing.unipi.it, luca.fanucci@unipi.it

Abstract—SpaceFibre is the forthcoming technology for very high-speed reliable data links in on-board satellite networks. Thanks to the evolution of detectors, each payload may generate data at very high rates, beyond 1 Gbps, for which there are no standardised network technologies available. SpaceFibre satisfies this need, making possible to reach data-rates higher than 2.5 Gbps per lane. Moreover, it also integrates a link-level Fault Detection Isolation and Recovery system and a Quality of Service mechanism using several Virtual Channels. While these features greatly simplify the upper layer applications, they also make undoubtedly complex for the system engineers to correctly configure the whole SpaceFibre network in order to achieve the desired end-to-end Quality of Service. In this paper, a Simulator for HIgh-speed Networks (SHINe) is presented. It has been developed to help the design of SpaceFibre and SpaceWire network architectures and to support the development and test of new upper layer protocols.

Index Terms— SpaceFibre, SpaceWire, Network Simulator, SHINe

I. INTRODUCTION

Science missions are requiring constantly increasing datarates, as demonstrated by missions like JUICE, MTG, CarbonSat, etc. Soon, the widely used SpaceWire will no longer be sufficient to satisfy the request for the data-rate. In addition, the growing complexity of on-board networks requires more and more features to the underlying interconnection technology, such as reliability, determinism and Quality of Service (QoS) capabilities.

SpaceFibre is a new high-speed standard, supported by the European Space Agency (ESA), integrating all these services directly in the network itself. Obviously, this makes the definition of the network architecture and the configuration of its QoS parameters a much more complex task than before. Moreover, a new Transaction Layer for SpaceFibre is currently under study and needs to be defined. The Transaction Layer will add commonly used services, such as a time distribution mechanism and an interrupt mechanism, to a SpaceFibre network.

David Jameux Space Research and Technology Centre European Space Agency Noordwijk, The Netherlands david.jameux@esa.int

The Simulator for HIgh-speed Networks (SHINe) is an easy-to-use software network simulator able to simulate both SpaceFibre and SpaceWire links and automatically derive network performances. SHINe is based on the widely used open-source framework OMNeT++, hence it is completely written as an object-oriented C++ library. It allows graphically or textually building a new network, connecting together existing building blocks (e.g., SpaceFibre Node, SpaceWire Node, Routing Switch Node). Each node can be individually configured to define its behaviour and the statistics to collect. SHINe has been developed as a highly modular tool, making possible to easily create new application protocols and to test them using the already existing SpaceFibre or SpaceWire building blocks.

SHINe is based on SpaceFibre Specification H8 [1] and on SpaceWire Specification 12C-Rev1 [2], with the noticeable exception that the SpaceWire Interrupt mechanism is not supported yet by the simulator. All the other features foreseen by the two standards have been implemented.

This paper is organized as follows:

Section II presents an overview of the already existing network simulators in this field.

Section III provides internal details of the SHINe software architecture.

Section IV shows an example of a simple network setup, briefly going through all the steps needed to define and simulate a network.

Finally, the conclusions are drawn in Section V.

II. RELATED WORKS

The most used software simulator for on board satellite networks comprising SpaceWire or SpaceFibre support is Modelling of SpaceWire Traffic (MOST) [3], developed by Thales Alenia Space (TAS) in collaboration with ESA. MOST is based on the toolkit OPNET, hence it is written in C language. While the use of MOST is free of charge in the context of ESA projects, OPNET still requires the payment of an annual fee to be used, even if recently the simulator has been partially ported to the new simulator engine NS3, which uses a GPLv2 license. In contrast, OMNeT++ is free of charge for research activities. To date, MOST supports the simulation of SpaceWire links, while the support for SpaceFibre is provided by an independent library that is available only for the OPNET version.

MOST provides a huge library of SpaceWire-related products. This includes not only "ideal" components, i.e. blocks whose behavior is derived only from the SpaceWire standard specifications, but also accurate models of commercialized products. Some examples are the SMCS116SpW, the SMCS332SpW, the RTC and the SpW-10X Switch. This library allows performing an accurate simulation of some real-case networks with a clock-cycle precision. MOST has already been used is some ESA projects, for example to test the SpaceWire network of the Meteosat Third Generation (MTG) mission [4].

While the long-term development of MOST brings some undoubted advantages, such as the availability of a wide components library, the high cost of OPNET and the currently lack of support for SpaceFibre in the NS3 version make SHINe an attractive alternative, in particular to study forthcoming networks requiring very high data-rates and QoS mechanism.

Other simulators have been developed in this field, such as SpaNSim [5] from Saint-Petersburg University and SpaceWire Simulator [6] from Mitsubishi. While the first is based on OPNET, the second relies on a mixed approach using both NS3 and FreeVHDL. To date, they do not support SpaceFibre.

III. SOFTWARE ARCHITECTURE

SHINe has been developed as a library for OMNeT++. It is realized as a highly modular collection of nodes, easily modifiable and extensible, making an extensive use of the object-oriented interface concept. In this section, the software architecture of the main building blocks available in the simulator is described.

A. SHINe Levels

The implementation of SpaceFibre and SpaceWire protocols is split into several levels, trying to represent the functional separation described in the two standards. Both SpaceFibre and SpaceWire share the same level division, although their implementation is different. The levels are:

- Endpoint Level;
- Port Level;
- Codec Level;

In general, the higher the level, the more abstract is the interface. A representation of the stack is depicted in Fig. 1.

The upper level, the closest to the application, is called Endpoint Level. It provides an NChar interface to the application to read and write data streams. Depending on the protocol implemented by the node, it also provides a Broadcast or a TimeCode interface. Note that in case of SpaceFibre, the NChar interface must distinguish among the Virtual Channels. The role of the Endpoint Level is mainly to collect statistics



Fig. 1 SHINe Level Architecture

and to handle TimeCodes in case of SpaceWire, or Broadcast messages carrying time information in case of SpaceFibre, validating or discarding them depending on their value. The Endpoint Level is connected to the Port Level, which still provides an NChar and Broadcast/TimeCode interface. The Port Level has the role to translate the unified NChars interface to the actual OMNeT++ message types, specific for SpaceFibre and SpaceWire, that are required by the Codec Level below. At Codec Level, any kind of abstraction is lost. The standard specifications are actually implemented in this level as C++ code.

Each level can be configured through many parameters. Examples of parametric values are: number of Virtual Channels (for SpaceFibre), logical address of the node, buffers size, etc. It is also possible to specify the delay in the transmission and reception path of each node, in order to simulate the presence of known delays in the data path (pipeline stages, elastic buffers).

It is important to note that Levels do not exchange messages using the OMNeT++ infrastructure, which would imply huge overhead and would increase the code complexity, but simply through function calls. The OMNeT++ messages are used only to interconnect the Codec Levels of different nodes. Hence, they are used only to simulate physical SpaceFibre or SpaceWire links.



Fig. 2 Architecture of a Routing Switch with three SpaceFibre ports and one SpaceWire port

B. SHINe Routing Switch

The core elements of a network are of course the Routing Switches, which allow interconnecting multiple nodes together. SHINe implements a Routing Switch able to handle three different kinds of ports: i) SpaceFibre ports; ii) SpaceWire ports; and iii) FIFO ports, which can be used to simulate a generic FIFO-like connection. The Routing Switch has been developed according to the SpaceFibre standard, which means that its SpaceWire ports need an adaptation layer to be connected to the Switching Matrix. A schematic representation is depicted in Fig. 2.

Thanks to the Adapter Port, the SpaceWire interface is seen by the Routing Switch as a SpaceFibre port with only one Virtual Channel. In addition, the adaptation layer translates the TimeCodes to Broadcast messages and vice-versa. When a TimeCode is received from a SpaceWire port, it is wrapped into a Broadcast message of a specific type and forwarded to the output ports (if this is the case, according to the SpaceFibre standard). If one of the output ports is a SpaceWire one, the Broadcast message is unwrapped and its type is checked. If the message contains a TimeCode it is forwarded, otherwise it is dropped. As result, the SpaceFibre network works as a backbone for SpaceWire TimeCodes. This solution offers a simple and intuitive abstraction for mixed networks.

The SHINe Routing Switch implements all the features foreseen by SpaceFibre, hence it supports:

- Both Path and Logical Addressing schemes;
- Optional header deletion in case of Logical Addressing;

- Virtual Network mapping, allowing to specify, for each Virtual Channel of each port, its Virtual Network number;
- Group Adaptive Routing;
- Multicast;
- Virtual Channel timeout;
- Broadcast logic;

The routing table used in case of Logical Addressing can be either provided by the user, in the form of a comma-separated file, or automatically built by the simulator itself using the Dijkstra's algorithm [7].

C. Test Applications

SHINe provides the building blocks to setup the network infrastructure, and usually it is up to the user to implement the specific application running on top of the network. However, in order to test generic network configurations, SHINe comes with two ready-to-use application nodes, TestAppSpfi and TestAppSpw, to be connected respectively to a SpaceFibre Endpoint and to a SpaceWire Endpoint.

TestAppSpfi can generate and consume data with a userdefined bandwidth on each Virtual Channel of the SpaceFibre Endpoint. It can also be used to generate and consume Broadcast messages. In a similar way, TestAppSpw can generate and consume data and TimeCodes from a SpaceWire Endpoint. In both cases, it is possible to specify the average value of the packet size as a Poisson distributions. This allows testing network performances under different load conditions. Both TestAppSpfi and TestAppSpw can be configured to use either Path or Logical addressing, without any change in the C++ code. Finally, they can perform a check on the content of the received packets, automatically informing the user in case an error occurred.

IV. NETWORK SETUP AND ANALYSIS OF THE RESULTS

This section shows an example of a simulation run, starting from the setup of the network up to the analysis of the results.

The first step is to define a new network in a .NED [8] file and populate it with the building blocks provided in SHINe, either graphically (through drag&drop) or textually. .NED is the file format used by OMNeT++ to describe both nodes and networks. The commonly used building blocks are: i) SpaceFibre or SpaceWire Endpoints; ii) Routing Switches; iii) User-defined applications connected to the SpaceFibre or SpaceWire Endpoints. In the following example, TestAppSpfi will be used in place of the user-defined application. The nodes must then be connected together using the appropriate links (SpaceFibre link, SpaceWire link or FIFO link in case of FIFO ports). Figure 3 shows an example of a possible network comprising one Routing Switch, four SpaceFibre Endpoints (Spfi0, Spfi1, Spfi2, Spfi3), and four TestAppSpfi (Gen0, Gen1, Gen2, Sink) attached to them.



Fig. 3 Example of a Network in SHINe

The second step is to configure the deployed nodes. This can be done in two different ways: directly modifying the .NED file of the network or using a separate configuration file (.ini file), which overwrites the parameters. Usually the first method is preferred for "hard" parameters, which are intrinsic of the network, such as the number of Virtual Channels of a certain node, while the second method is better suited for the parameters that can vary from run to run, such as the rate at which an application produces or consumes data. The simulation in Fig. 3 is configured to have four Virtual Channels per port, with three TestAppSpfi used to generate data (Gen0, Gen1 and Gen2) and one used to consume data (Sink). All SpaceFibre Codecs in the network have their Expected Bandwidth parameter set to [10%, 20%, 30%, 35%]. The Expected Bandwidth parameter specifies the percentage of the link utilization assigned to each Virtual Channel. They sum up to 95% to take into account the protocol overhead required for Flow Control Tokens (FCTs) and ACK/NACK words.

Once the network is fully configured, the simulation is ready to run. OMNeT++ allows running the simulation either graphically, showing the data exchanged among the nodes, or completely through command line. When the graphical interface is used, it is also possible to pause the simulation and deeply inspect the status of each node, which can be useful for debugging purposes.

At the end of the simulation, the tool creates a file containing the performance results to be analysed. Many core metrics are measured by default by SHINe, such as packets latency, bandwidths, Flow Control Tokens (FCTs) counters, etc. The user can also independently turn them off in order to speed up the simulation. OMNeT++ offers out-of-the-box a powerful graphical tool to explore, filter and process the



Fig. 4 Link utilisation percentage per Virtual Channel

results, together with the possibility to export them for postprocessing with external tools (Octave, MATLAB).

Continuing the example described in Fig. 3, one of the possible metrics that is possible to analyse is the Link utilisation per Virtual Channel, to check if it is in line with the expected one.

Figure 4 shows one of the possible results obtained running the example described above, in which the percentage utilisation of the total link bandwidth is shown for each Virtual Channel of each source application. Because the three applications generating data have to share the same output port in the Routing Switch, they cannot transmit at full bandwidth but roughly at one third. In this scenario, the link utilisation per Virtual Channel reflects the proportion of the Expected Bandwidth parameter previously set, considering that each transmitting node can use up to one third of the total link bandwidth (without taking the protocol overhead into account). The oscillation is due to the wormhole routing mechanism used to forward packets to the output port of the Routing Switch. Indeed, when a transmitting node obtains the output port assigned for one of its Virtual Channels, the other nodes cannot transmit on the same Virtual Channel. The result is that the transmission happens in bursts, whose length depends on the size of the packets. In this experiment, the packet size has been fixed to 25 words, producing very small bursts and a fair utilisation of the output port, hence relatively small oscillations around the mean value. It is important to note that the results given by SHINe can also be used to validate the behaviour of a real device, comparing their respective results, if clock-cycle precision is not needed.

V. CONCLUSIONS

The SHINe network simulator has been described in this paper. SHINe is a flexible tool, entirely based on the open-

source framework OMNeT++, which can be used to rapidly setup SpaceFibre and SpaceWire networks by deploying and interconnecting both existing and new user-defined nodes. Among the others, the main building blocks provided by SHINe are i) a SpaceFibre node, ii) a SpaceWire node, iii) a SpaceFibre Routing Switch, iv) a SpaceWire-to-SpaceFibre adapter making possible to connect SpaceWire ports to the Routing Switch. Thanks to these nodes, the user can create and test mixed networks. The Routing Switch features all the advanced functionalities foreseen by the standard, including Multicast, Group Adaptive Routing, Broadcast message forwarding and a completely configurable Virtual Channel to Virtual Network mapping. SHINe also provides some ready-touse application nodes that can be used to test the network under different load conditions. Given the increasing need for a highspeed network technology in space missions, SHINe proves itself a flexible tool to simulate performances, possibly usable to validate the behaviour of hardware devices, and to help the development of future upper layer protocols for SpaceFibre and SpaceWire networks.

REFERENCES

- S. Parkes, A. Ferrer, A. Gonzalez, C. McClements, "SpaceFibre Specification Draft H8", University of Dundee, January 2017
- [2] "ECSS-E-ST-50-12C Rev.1 DIR3", ESA-ESTEC, November 2015
- [3] Brice Dellandrea, Baptiste Gouin, Steve Parkes, David Jameux, "MOST: Modelling of SpaceWire and SpaceFibre Traffic Applications and Operations: On-Board Segment", Thales Alenia Space, 2014
- [4] B. Dellandrea, "MOST MTG-I Simulation Presentation"
- [5] Artur Eganyan, Liudmila Koblyakova, Elena Suvorova, "SpaceWire Network Simulator", Saint-Petersburg University of Aerospace Instrumentation
- [6] Michiya Hayama, Yosuke Yokoyama, Riko Yagiu, Isao Odagi, Hiroto Namikoshi, "Impacts of Faults on a SpaceWire Network", Mitsubishi Electric Corporation
- [7] https://en.wikipedia.org/wiki/Dijkstra%27s_algorithm
- [8] https://www.omnetpp.org/doc/omnetpp/manual/

Standardization Efforts for a Network Management and Discovery Protocol for SpaceFibre

Standardization (Short)

Felix Siegle On-board Payload Data Processing Section European Space Agency – ESTEC Noordwijk, The Netherlands spacefibre@esa.int

Abstract — We present a proposal for a unified configuration (and status) memory space for SpaceFibre devices that simplifies the management of complex networks and allows automatic node discovery. Aside from mapping all SpaceFibre port status and configuration parameters as specified in the SpaceFibre standard to corresponding registers, it also contains parameters for routing switches that we believe are important to be standardized as well. In addition, definitions of broadcast status messages are proposed, which can automatically be generated and transmitted by routing switches in case of failures or anomalies within the network.

Index Terms - Network Configuration, Network Discovery SpaceFibre, Standardization

I. INTRODUCTION

SpaceFibre is the new European high-speed serial data link and the successor of the very popular SpaceWire protocol. It is specifically designed for spaceflight applications and incorporates several Quality-of-Service (QoS) techniques into its hardware. A single network link combines several data streams by means of virtual channels. They are multiplexed onto the link based on reserved bandwidth, priorities, time-slots, or a combination of these mechanisms. Integrated Fault Detection, Isolation and Recovery (FDIR) techniques guarantee fault-free communication on a point-to-point basis. Comparable to other modern intercommunication architectures such as Serial RapidIO, Serial ATA or PCI Express, SpaceFibre communicates over a Serializer/Deserializer (SerDes) device and can thus reach data signaling rates of several Gigabits per Second (Gbps). In addition, the rates can be increased by using several SerDes lanes in a multi-lane configuration. Due to its native support of the SpaceWire packet format, its small area overhead and its high performance even on smaller space-qualified Field Programmable Gate Arrays (FPGAs), SpaceFibre is particularly well suited for future high-speed on-board communication.

At the time of writing the SpaceFibre standard ECSS-E-ST-50-11C [1] is in public review before being finally released by the European Cooperation for Space Standardization (ECSS). It specifies a SpaceFibre port (the actual network interface comprising the data link, multi-lane, lane and physical layer) in detail and the functionality of a SpaceFibre network (the Alessandro Leoni Department of Information Engineering Università di Pisa Pisa, Italy alessandro.leoni@ing.unipi.it

network layer) in a more general way. Therefore, the standard is perfectly suited to implement SpaceFibre ports, point-to-point connections as well as simple networks. However, extra functionality might be required to make also more complex and larger networks manageable. For this reason, the European Space Agency established a working group that deals with the development of an upper protocol, which in the following will be called *transaction layer*.

This paper gives an overview of the initial idea and the components of this transaction layer. First, the scope of the transaction layer is given in Section II. Then, a draft of one part of the transaction layer, the unified configuration space, is explained in Section III. A possible definition of broadcasts that are used for signaling error conditions is given in Section IV before the paper is concluded in Section V.

II. SCOPE OF THE TRANSACTION LAYER

In an early assessment, the following elements of the transaction layer have been identified:

- Unified mechanisms for network administration: One unified configuration space for SpaceFibre routing switches and end nodes including means to identify the device during network discovery and management.
- *Time synchronization*: Synchronizing the time of all nodes in the network by sending broadcast messages is required for the QoS mechanisms on both the data link layer (the SpaceFibre time-slots) and the application layer. For this reason, the transaction layer must define algorithms to distribute and possibly adjust the time, it must define the format of the broadcast messages, and it must investigate redundancy concepts of multiple time masters.
- *End-to-end QoS:* The transaction layer must standardize how the scheduling on the application layer maps to the mechanisms built into the data link layer (the SpaceFibre time-slots). It could also define an end-to-end flow control protocol.
- Fault Detection, Isolation and Recovery: A Failure Mode and Effects Analysis (FMEA) should reveal

which possible failures and anomalies can occur in a SpaceFibre network. From these results, the following elements can be specified: (i) mechanisms to detect and isolate errors, (ii) status flags and other status information that indicate these error conditions, (iii) a unified protocol based on broadcast messages to report errors from the network to the network manager (which is certainly the on-board computer), and (iv) a unified protocol based on broadcast messages that allows the network manager to interact with and control the network.

- SpaceWire to SpaceFibre bridging and vice-versa: The transaction layer must specify how SpaceWire timecodes are mapped to SpaceFibre broadcast messages. It must also discuss how SpaceFibre virtual networks are mapped to SpaceWire ports within a mixed routing switch and the possible consequences in terms of QoS.
- *Definition of roles in a multi-port design:* It must be clarified which port(s) in a multi-port end node are responsible for the time synchronization and the reception of configuration packets via virtual network 0.

Service interfaces for the network manager can be defined once all these elements are specified. In practice, this could lead to a software Application Programming Interface (API) that can be integrated into an on-board computer program to execute the initial configuration of the network, to monitor the health status of the network, and to interact with the network in case of failures.

III. UNIFIED CONFIGURATION SPACE

A. Overview

As a first step, a unified configuration space for both SpaceFibre routing switches and end nodes has been drafted with the goal to have the same configuration interface in every future SpaceFibre device, independently of the manufacturer. The current SpaceFibre standard defines that the Remote Memory Access Protocol (RMAP) [2] shall be used for remote configuration, control and monitoring of SpaceFibre networks. The access is only allowed via virtual network 0, which is always fixed to virtual channel 0 of a SpaceFibre port. In practice, a simple RMAP Intellectual Property (IP) core can be connected to virtual channel 0 that maps the unified address space to internal configuration and status registers. The RMAP protocol is based on byte-wise addressing and the same approach is followed for the SpaceFibre configuration space although other alignments are possible too. In total, the width of the address space is 24 bits.

B. The Global Selector

The global selector at address bits [23:22] is used to distinguish between four subspaces, as can be seen in Table I.

TABLE I. GLOBAL SELECTOR

A[23:22]	A[21:17]	A[16:0] Subspace
00	Reserved	<i>Node configuration space</i> : This subspace comprises the general fields related to all nodes (end nodes and routing switches) as well as fields specific to only routing switches.
01	Number of ports	SpaceFibre port configuration space: This subspace comprises the fields related to the SpaceFibre port(s).
10	Manufacturer configuration space: This space is used by the manufacturer to extend the status and configuration fields by fields specific to this device. The network manager can be made aware of the meaning of these fields with the help of Electronic Data Sheets.	
11	<i>User configuration space:</i> The configuration space can be further extended by the user. For instance, a scientific instrument using SpaceFibre could add fields for the configuration of the instrument itself. Again, by means of Electronic Data Sheets, the network manager could be made aware not only of the connected SpaceFibre device but also of the actual application behind it.	

C. The Node Configuration Space

The node configuration space has two functions. Firstly, it comprises fields that are required for network discovery. These fields are used both for end nodes and for routing switches. Secondly, it comprises fields specifically dealing with the routing switches. The address bits [16:15] define the sub-blocks as listed in Table II. Except of the first block, the content of the other blocks repeats depending of their meaning. Table III lists all fields of the general block, Table IV the fields of the routing table block (shown only for one routing table entry), and Table V the fields of the virtual channel block (shown only for one virtual channel).

T / D / D / I

A[16:15]	A[14:0] Sub-Block		
00	<i>General block:</i> Fields related to both end nodes and routing switches. This block with general status and configuration does not repeat and is mandatory.		
01	<i>Routing table block:</i> Fields related to the configuration of the routing tables. A routing table has at least as many entries as logical addresses (255-32) and therefore the content of this block repeats just as often. Which logical address is currently accessed is defined by the address bits [11:4].		
10	<i>Virtual channel block:</i> Fields related to each virtual channel of each port. The configuration space supports up to 32 ports, the currently accessed one is defined by the address bits [14:10]. Per port, it supports up to 32 virtual channels and the currently accessed one is defined by address bits [9:5].		
11	Reserved or used as additional manufacturer fields.		

TABLE III. NODE CONFIGURATION: GENERAL BLOCK

A[3:0]	Field	Description
0x0	Static Config. 1	Bits [7:6]: Extends the class ID. Bits [5:0]: Describes the class of the device, e.g. end node or routing switch.
0x1	Static Config. 2	Product ID that describes the specific
0x2	(16 bits)	product. This ID could be assigned

A[3:0]	Field	Description
		and managed by the European Space
		Agency.
		Node ID that is dynamically assigned
0x3	Configuration 1	by the network manager after
0x4	(16 bits)	discovery and is subsequently used to
		identify the device on the network.
	Configuration 2	Broadcast Channel number used by
0x5		this device. It is dynamically assigned
		by the network manager.
0x6 –	Configuration 3	Timeout parameter for the
0x9	(32 bits)	propagation of broadcast messages.
0	Statia Config 2	Bits[4:0]: The number of SpaceFibre
OXA Static Config. 3	Static Coning. 5	ports in this device.
0xB	Statia Config 1	Bits[4:0]: The number of SpaceWire
	Static Config. 4	ports in this device.
0xC	Statia Config 5	Bits[4:0]: The number of FIFO ports
	Static Config. 5	in this device.

TABLE IV. NODE CONFIGURATION: ROUTING TABLE BLOCK

A[3:0]	Field	Description
0x0 - 0x3	Configuration 1 (32 bits)	Selected output port(s). For each logical address, one or more output ports can be chosen by setting the corresponding bit to 1.
0x4 – 0x7	Configuration 2 (32 bits)	Remove header. The header of the packet is removed when it is routed to the output port(s) set to 1 in this field.
0x8	Configuration 3	Bit[0]: Multicast. If set and if several ports are selected, the packet is multicast to the selected ports. If not set, group adaptive routing is used.

 $TABLE \ V. \ \ NODE \ CONFIGURATION: \ VIRTUAL \ CHANNEL \ BLOCK$

A[4:0]	Field	Description
0x0	Configuration 1	Bits[5:0]: This field is used to assign the virtual channel to a specific virtual network.
0x1	Interrupt Enable	Bits[4:0]: If a bit is set, an interrupt broadcast message is generated for the events listed in the Error Flags field.
0x2	Error Flags	Bit[4]: Timeslot guardian violation Bit[3]: Wormhole timeout expired Bit[2]: V. channel timeout expired Bit[1]: Virtual network decode error Bit[0]: Routing table decode error
0x3	Status 1	Routing table decode error: stores the address that caused the error.
0x4	Status 2	Bits[4:0]: Virtual network decode error: stores the requested output port that led to the error.
0x5	Status 3	Bits[5:0]: Virtual network decode error: stores the affected virtual network number.
0x6 – 0x9	Configuration 2 (32 bits)	Virtual Channel Timeout. Stores the maximum timeout value associated with this virtual channel.
0xA – 0xD	Configuration 3 (32 bits)	Wormhole Timeout. Stores the maximum time a wormhole is allowed to stay open in this virtual channel.
0xE – 0x15	Configuration 4 (64 bits)	Time-slot Guardian. The time-slot guardian checks if a virtual channel is communicating outside of its assigned time-slot (selected by setting the corresponding bit to 1).

A[4:0]	Field	Description
0x16	Configuration 5	Bit[3]: Cancel current packet transfer in case of a timeout Bit[2]: Enable timeslot guardian Bit[1]: Enable wormhole timeout Bit[0]: Use virtual channel timeout

D. The SpaceFibre Port Configuration Space

The SpaceFibre port configuration space is used for the configuration and status information of each SpaceFibre port in the system. It consists of three blocks selected by address bits [10:9] as listed in Table VI. The content of the first general block (shown in Table VII) does not repeat, the content of the second block repeats for each lane of the SpaceFibre port (shown for only one in Table VIII) and the content of the third block repeats for each virtual channel of the SpaceFibre port (shown for only one in Table IX).

TABLE VI. PORT CONFIGURATION SPACE SELECTOR

A[10:9]	A[8:0] Sub-Block	
00	<i>General block:</i> Fields related to the SpaceFibre port in general.	
01	<i>Lane block:</i> Fields related to the lane(s) of the SpaceFibre port. Bits[6:3] define the number of the currently addressed lane.	
10	<i>Virtual channel block:</i> Fields related to the virtual channel(s) of the SpaceFibre port. Bits [8:4] define the number of the currently addressed virtual channel.	
11	Reserved or used as additional manufacturer fields.	

TABLE VII. PORT CONFIGURATION: GENERAL BLOCK

A[4:0]	Field	Description
0x0	Static Config. 1	Bits[7:6]: Reserved Bits[4:0]: Indicates how many virtual channels are available in this SpaceFibre port.
0x1	Static Config. 2	Bits[3:0]: Indicates how many lanes are available in this SpaceFibre port.
0x2 0x3	-	Reserved
0x4 0x5	Static Config. 3 (16 bits)	Indicates which lanes are configured to send data.
0x6 0x7	Static Config. 4 (16 bits)	Indicates which lanes are configured to receive data.
0x8	Interrupt Enable	Bits[5:0]: If a bit is set, an interrupt broadcast message is generated for the events listed in the Error Flags field.
0x9	Error Flags	Bit[5]: Far-end link reset Bit[4]: Link reset caused by protocol Bit[3]: Sequence error Bit[2]: CRC-8 error Bit[1]: Frame error Bit[0]: CRC-16 error
0xA	Status 1	Bits[5:0]: Indicates the number of error recovery attempts.
0xB	Status 2	Bits[2:1]: Alignment state Bit[0]: Error Recovery Buffer Empty
0xC	Configuration 1	Bits[7:3]: Maximum number of data sending lanes. Bit[2]: Interface reset Bit[1]: Link reset Bit[0]: Scrambler

A[4:0]	Field	Description
0xD	Configuration 2	Bits[6:0]: Configures the expected broadcast bandwidth value.
0xE 0xF	Configuration 3 (16 bits)	Configures the virtual channel idle time limit.
0x10 - 0x13	Configuration 4 (32 bits)	Configures the bandwidth credit limit.
0x14	Configuration 5	Bit[7]: Write to clear the number of error recovery attempts. Bits[6:3]: Data segment multiplier. This value is used to configure the length of one data segment. Bits[2:0]: FCT multiplier. This value is used to configure how much data is covered per FCT.

TABLE VIII. PORT CONFIGURATION: LANE BLOCK

A[2:0]	Field	Description						
		If a bit is set, an interrupt broadcast						
		messages is generated for the						
0×0	Interrupt Enable 1	following events:						
UAU	Interrupt Enable 1	Bit[4]: No signal						
		Bit[1]: Lane state: Clear-line						
		Bit[0]: Lane state: Active						
		Bit[6]: Bit synchronization						
0x1	Status 1	Bit[5]: RX polarity						
UAI	Status I	Bit[4]: No signal						
		Bits[3:0]: Lane state						
0x2	Status 2	Indicates the value of the RX error						
UAL	Status 2	counter.						
		If a bit is set, an interrupt broadcast						
		message is generated for the						
		following events:						
0x3	Interrupt Enable 2	Bit[3]: Far-end lost signal						
		Bit[2]: Timeout						
		Bit[1]: Far-end standby						
		Bit[0]: RXERR counter overflow						
		Bits[7:4]: Far-end capabilities						
		Bit[3]: Far-end lost signal						
0x4	Status 3	Bit[2]: Timeout						
		Bit[1]: Far-end standby						
		Bit[0]: RXERR counter overflow						
		Bit[7]: Far-end serial loopback						
		Bit[6]: Near-end serial loopback						
0x5		Bit[5]: Parallel loopback						
	Configuration 1	Bit[4]: Lane reset (write clear)						
	Configuration	Bit[3]: Auto-start						
		Bit[2]: Start mode						
		Bit[1]: TX enable						
		Bit[0]: RX Enable						
0v6	Configuration 2	Bits[3:0]: Configures the standby						
0x6		reason						

TABLE IX. PORT CONFIGURATION: VIRTUAL CHANNEL B	LOCK
---	------

A[3:0]	Field	Description
0x0	Interrupt Enable	If a bit is set, an interrupt broadcast messages is generated for the following events: Bit[3]: FCT credit counter overflow Bit[2]: Input buffer overflow Bit[1]: Bandwidth underuse Bit[0]: Bandwidth overuse
0x1	Status	Bit[4]: Virtual channel has credit Bit[3]: FCT credit counter overflow Bit[2]: Input buffer overflow Bit[1]: Bandwidth underuse Bit[0]: Bandwidth overuse

A[3:0]	Field	Description
0x2	Configuration 1	Bits[4:0]: Priority level
0x3	Configuration 2	Bits[6:0]: Expected bandwidth
0x4 – 0xB	Configuration 3 (64 bits)	Assigned timeslots

IV. DEFINITION OF BROADCASTS

As seen in Table V, VII, VIII, and IX, interrupts can be enabled for various (error) conditions occurring in either the routing switch or one of the ports. The interrupt generation triggers the creation and transmission of a broadcast code that can be used by the network manager (or any other node on the network) to monitor the health status of the network with very quick update rates. Since a broadcast can have a payload of 64 bits, it is not required to define one for each interrupt. Instead, status information is logically grouped into four types of broadcast messages: (i) related to the routing switch, (ii) related to a SpaceFibre port in general, (iii) related to a specific lane of a SpaceFibre port, and (iv) related to a specific virtual channel of a SpaceFibre port. Since the Node ID is also part of the broadcast transmission, the source device is immediately known to the network manager. The content of each broadcast message type is listed in Tables X – XIII.

TABLE X. STATUS BROADCAST: ROUTING SWITCH

Payload Word	Description				
	Bits[28:24]: Number of affected virtual channel				
Word 1 [63:32]	Bits[20:16]: Number of affected port				
	Bits[15:0]: Node ID of source device				
	Bits[29:24]: VN decode error: virtual network				
	Bits[20:16]: VN decode error: output port				
	Bits[15:8]: Routing table decode error: address				
Word 2 [21:0]	Bit[4]: Timeslot guardian violation				
wolu 2 [51.0]	Bit[3]: Wormhole timeout expired				
	Bit[2]: Virtual channel timeout expired				
	Bit[1]: Virtual network decode error				
	Bit[0]: Routing table decode error				

TABLE XI. S	TATUS BROADCAST: SPACEFIBRE PC	ORT GENERAL

Payload Word	Description				
Word 1 [62:22]	Bits[20:16]: Number of affected port				
wolu 1 [05.52]	Bits[15:0]: Node ID of source device				
	Bits[18:17]: Alignment state				
	Bit[16]: Error Recovery Buffer Empty				
	Bits[13:8]: Number of error recovery attempts				
	Bit[5]: Far-end link reset				
Word 2 [31:0]	Bit[4]: Link reset cause by protocol				
	Bit[3]: Sequence error				
	Bit[2]: CRC-8 error				
	Bit[1]: Frame error				
	Bit[0]: CRC-16 error				

Payload Word	Description
	Bits[27:24]: Number of affected lane
Word 1 [63:32]	Bits[20:16]: Number of affected port
	Bits[15:0]: Node ID of source device
	Bits[23:20]: Far-end capability
Word 2 [31:0]	Bit[19]: Far-end lost signal
	Bit[18]: Timeout

Payload Word	Description
	Bit[17]: Far-end standby
	Bit[16]: RX error overflow
	Bits[15:8]: RX error counter
	Bit[6]: Bit synchronization
	Bit[5]: RX polarity
	Bit[4]: No signal
	Bit[3:0]: Lane state

TABLE XIII. STATUS BROADCAST: PORT VIRTUAL CHANNEL

Payload Word	Description				
	Bits[28:24]: Number of affected virtual channel				
Word 1 [63:32]	Bits[20:16]: Number of affected port				
	Bits[15:0]: Node ID of source device				
	Bit[4]: FCT credit counter overflow				
	Bit[3]: Input buffer overflow				
Word 2 [31:0]	Bit[2]: Virtual channel has credit				
	Bit[1]: Bandwidth underuse				
	Bit[0]: Bandwidth overuse				

V. CONCLUSIONS

In this paper, a novel configuration (and status) memory space has been presented that could easily be integrated into every future SpaceFibre routing switch and end node device. By doing so, a network manager can automatically discover nodes in the network, read the essential status information from them, and configure their essential SpaceFibre parameters. In addition, the memory map leaves enough space also for manufacturer and user registers. The status information available from SpaceFibre routing switches and end nodes has also been mapped to four status broadcast messages that can automatically be generated in case of failures within the network. As a next step, it will be required to also define configuration broadcast messages that can be used by a network manager to interact with the components of the SpaceFibre network.

ACKNOWLEDGMENT

The European Space Agency, the University of Pisa and IngeniArs S.r.l. have partly funded this work under the NPI program.

REFERENCES

- [1] European Cooperation for Space Standardization, SpaceFibre Very high-speed serial link, ECSS-E-ST-50-11C DIR1, Noordwijk: ECSS Secretariat, 2018.
- [2] European Cooperation for Space Standardization, SpaceWire Remote Memory Access Protocol, ECSS-E-ST-50-52C, Noordwijk: ECSS Secretariat, 2010.

Poster Presentations

SpaceWire Meets Big Data - Realtime Data Mining

SpaceWire Networks and Protocols, Short Paper

Thomas Bahls, Markus Bihler and Sergey Tarassenko Institute of Robotics and Mechatronics German Aerospace Center (DLR) Münchnerstr. 20 82234 Weßling, Germany Thomas.Bahls@dlr.de

Abstract—The whole is more than the sum of its parts. This saying particularly fits in the context of big data and data mining. A single data set is only valid and relevant at exactly one point in time. Dependencies between the individual elements of the data set as well as the behaviour over time can only be investigated if large amounts of data are available. A good example is traffic monitoring based on GPS data, WiFi, and/or cellular triangulation of modern mobile phones (see [1], [2]). Here, the data sets are collected from multiple sources to estimate a precise traffic situation.

Data completeness is vital to achieving sufficient reflection of the actual state of a system, since information can be lost through downsampling. This is a difficult requirement to fulfill especially for highly dynamic systems like robots. Data recording over the same communication infrastructure from multiple sources must meet strict real time requirements of the control application. Main control loops at joint level of $\geq 3 \text{ kHz}$ and up to 100 kHz at motor control are quite common (e.g. [3]). Due to this, a simple duplication of the cyclic data sets at CPU-level for simultaneously controlling and recording is not possible without decimation of the recording sample rate. Therefore recording has to be performed at a different point of the communication infrastructure. This approach provides a duplication of the data at SpaceWire packet level by using seperate SpaceWire regions for control and recording without violating the real time requirements of the control application.

I. INTRODUCTION

Today's high speed communication in combination with ever increasing data storage capabilities enables the recording of huge amounts of data. This has opened a new field of research, since in addition to a single data set of a system at a unique point in time, the system behaviour over time can also be taken into consideration. This enables the investigation of the relations between the different parameters among each other by using e.g. statistical methods. Such data logging and evaluation is often referred as *Big data* and *Data mining*. The prime example, used daily by millions of people, is the real time monitoring of traffic based on mobile GPS devices.

In a more specific context, big data can bring forward new solutions and possibilities. For robotic systems, the analysis of all arising data can help in many applications such as error tracking, temperature drift compensation, calibration, machine learning. However, there are strict requirements and constraints for data logging in robotic systems. In the example of traffic monitoring of the previous paragraph, collecting the required information is a decentralized task. Communication delays –



Fig. 1. DLR MirSurge System with three SpaceWire based DLR Miro arms and two SpaceWire based DLR Mica instruements, is the target platform for data recording.

even of a few seconds - do not matter. Several wrong data sets are also acceptable. For robotics, high data rates and real time communication are concentrated in a small area. Transmitted data in the robot's network has to be recorded without affecting the communication's determinism, to guarantee the desired control loop frequency. This requires a flexible and transparent communication, accessible on each layer and at each point in the network. Commercially available robots lack the required transparency and flexibility to implement such a data logging concept. On the one hand, it is often not possible to access raw data or interpret it without in-house knowledge. Additionally, logging raw data of an encapsulated robot inserts additional delay, which affects the determinism of the robot. On the other hand, data logging on a higher abstraction layer is possible, but only over a short period of time, or would require downsampling of the main control loop (see [4] [5]).

At the Institute of Robotics and Mechatronics of the German Aero Space Center (DLR) several robots employ SpaceWire as the communication backbone. Due to its properties, SpaceWire fits well to robotic demands (see [6], [7], [8], [3], [9]). It has a small footprint, supports arbitrary network topologies and is easy to implement [10]. In combination with an adapted physical layer [11] it also supports high bandwidth and low latency, which is essential for high dynamic robots to enable main control loops ≥ 3 KHz. Furthermore the SpaceWire

communication of these robots is completely FPGA based owing to this also the accessibility at each layer and each point in the network is guaranteed. Hence the requirements for big data are also given.

This paper presents an approach for SpaceWire raw data logging at main control loop frequency without any influence to the system's determinism. In section II a brief overview of the two approaches for data logging (serial and parallel) is given. Section III describes the chosen concept. The results based on our three arm medical robot setup (see Fig. 1) are presented in chapter IV. The paper concludes in section V and gives an outlook in section IV of future work.

II. STATE OF THE ART

To enable data logging in a network, data recording at an arbitrary point has to be provided. There are two different methods for data recording: parallel and serial. In case of parallel data recording a data recorder (DR) does not insert any delay into the communication channel. Figure 2 depicts this behaviour. The two nodes N1 and N2 are connected via an arbitrary network topology (sequence of links and routers) depicted as black lines. At a specific point data is recorded by simply observing and buffering of the data without adding any delay to the communication channel. A typical example is a PCI / PCI-Express analyzer.



Fig. 2. Parallel data recording

Conventional solution for the parallel data recording requires additional wiring and sophisticated electronics to cope with the high data rates and frequencies. Their size and complexity make it impossible to integrate them into a robotic system, especially for the point of data logging at an arbitrary point in the network. A possibility to overcome this problem is serial data recording (see Fig. 3). Here, an extra device is inserted at an accessible point of the network where the desired data are available (e.g. directly at the interface to the real time host).



Fig. 3. Serial data recording

The disadvantage is the inserted timing delay and jitter due to additional hardware in the communication channel, which is often not acceptable due to strong realtime requirements of high dynamic robotic systems.

III. CONCEPT

To avoid a delay in the communication path due to extra hardware, parallel data recording has to be considered directly at the design phase of the robotic system by the design of the network topology to provide routing capabilities for both control and data recording. As mentioned previously SpaceWire fits well to the robotic demands as well as to the requirements given by big data and data mining. Additionally to the accessibility at each layer and at arbitrary points the FPGA based SpaceWire communication also enables the integration of data recording units directly at specific points in the FPGA.

Taking continuous long term calibration and anomaly detection as an example for big data to optimize the robot's accuracy and reliability, all outgoing and incoming data have to be recorded. In this way a LUT comprising the full temperature range in different joint configurations can be built. Therefore, the appropriate point in the network is in the SpaceWire-HS (high speed) host adapter (see [12]), which serves as the interface from SpaceWire to the standard CPU of the real time host. Figure 4 shows its network topology.



Fig. 4. Network topology inside the SpaceWire-HS adapter

Inside the SpaceWire-HS host adapter are two separate regions: One for control (see lower area of Fig. 4 with grey background), and the other one for recording (see upper area of Fig. 4 with grey background). The routing switch (RS) separates these two regions (see dashed lines in Fig. 4). With the routing switch implemented in crossbar architecture interaction between the two region is avoided since exclusive routing resources are provided for both regions. This guarantees full bandwidth for control and recording since all packet routing inside one region is independent from the other.

Seven of the ten ports are part of the control region. Ports four, five, and six are exchange level implementations, which can be connected to spatially separated SpaceWire hardware like for example the joint electronics of the DLR Miro robot [7]. N2, N3 and N8 are request response targets according to [9]. These nodes are not part of the cyclic real time communication and have only housekeeping and configuration functionality of the SpaceWire-HS adapter. The node N1 serves both as a datagram sink and source implemented in software on the real time host (Linux extended by the real time patch PREEMPT-RT¹), and is the destination and origin for all control messages. Its connection to the routing switch is the point of interest since all control data passes this channel.

Both the transmit and the receive channel of N1's routing switch connection are observed and buffered by DRs. The DRs are directly connected to the nodes N9 and N10, which are implemented as datagram sources according to [9] inside the recording region. The data recorder buffering is not limited to the cargo. The SpaceWire address and transport level specific extensions (transport id and CRC, see [9]) are also stored. This is beneficial for debugging. Each DR is able to store the complete data traffic of a control cycle for redundancy. This prevents the loss of data in case of a sporadic delay in the recording path due to the determinism of the recording host target. The datagram sources N9 and N10 are triggered if a complete packet is buffered. N9 and N10 use the same exchange level implementation connected to port seven (see 4) to transmit there data to the recording host (a Linux PC). However the two datagram sources do not pose a conflict since in the control application the desired data are the result of the control model based on the last actual data. Hence the transmit and receive channel of N1 are never used simultaneously. The recording host is also equipped with SpaceWire-HS adapter, and is responsible for long term storage.

The recording host first stores all received raw packets during runtime into a single file. This file is divided into separate files offline regarding the destination address. Finally, these files are converted into a format readable for the control application. To have a full image of the system states, the Ethernet based communication of the different spatially distributed control models are recorded simultaneously by the same host. Thus the complete behaviour over the recorded period of time is traceable.

IV. RESULTS

The presented concept is evaluated using the SpaceWire equiped medical robot, DLR Miro [7] and its associated instrument the DLR Mica [8]. The DLR Miro is a seven degree of freedom lightweight robot designed for conventional and minimally invasive surgical applications. With its integrated force torque sensing, it provides several control modes which enable also the close interaction with medical staff like e.g. collision detection mode or hands on mode (manual guiding of the robot). Mountable to the DLR Miro's hollow shaft is the DLR Mica. It is a three degree of freedom instrument following the same design approach like the DLR Miro. The Mica can be equipped with different surgical tools (gripper, scissors, ...). Both the Miro and Mica have main control loops on joint level of 3 KHz. The cyclic data of the DLR Miro comes to around 850 Bytes and of the DLR Mica around 120 Bytes. The SpaceWire-HS host adapter and the network topology are use for the interface to the standard CPUs of both systems as presented in section III. The data recording is also performed by a standard Linux PC equipped with the SpaceWire-HS host adapter. The following test scenarios are performed:

- Long term data recording of the DLR Mica. During a demonstration application on a fair the complete data transfer of the DLR Mica was recorded over 20 h. The recording functionality was available incessantly over the whole runtime of the DLR Mica.
- Long term data recording of the DLR Miro. During teleoperation manipulation tasks of our three arm system the complete data transfer of a single DLR Miro was recorded over several sessions (each 2 h-3 h). The higher amount of data compared to the DLR Mica was fully manageable by the recording host. Recording functionally was constantly available for the whole period of time.
- Short term data recording for debugging of sporadically incurred system failures: Despite our medical robotic system's high level of reliability, sporadic non reproducible system failures happen from time to time. To eliminate these remaining failures a short term data recording, which only comprises a window of a few seconds, was implemented. This enables detailed examination of the failure and its cause. This feature was tested in different scenarios and can be activated on demand for both the DLR Miro and the DLR Mica.
- Simultaneously long term data recording of three DLR Miros: To test the recording host's capability the three arm scenario described in [13] was utilized. Therefore the data of three DLR Miro robots were simultaneously recorded over several hours without loss of data.

As a consequence of these successful tests we also integrated this approach on our eight-port switch, which is used for interconnection of regions to built large SpaceWire networks comprised of multiple systems. Initially the eight-port switch uses internally a ten-port SpaceWire routing switch. Eight ports with an exchange level implementation are provided for interconnection. The remaining two ports are used internally for housekeeping of the device and to update its firmware.

The new approach of the eight-port switch internally uses a 26-port SpaceWire routing switch (an extract is depicted in Fig. 5). Ports 1 to 8 with the exchange level implementation for interconnections, as well as the ports 9 and 10 for housekeeping and firmware updating (see Fig. 5 for N9 and N10) are identical. The ports 11 to 26 are used for recording the raw data packets on the transmit and receive channels of the eight exchange level implementation. Fig 5 illustrates this for

¹Information about the PPEEMPT-RT patch under https://rt.wiki.kernel.org



Fig. 5. Network topology inside the eight port SpaceWire switch with data recording functionality

one exchange level implementation with its data recorders and the associated diagram sources N11 and N12. The remaining seven exchange level implementations are duplication of this (shown in the dashed frame of Fig. 5).

With this new approach of the eight-port switch the routing capabilities can be individually configured regarding recording and control according to the actual system and application requirements.

V. CONCLUSION

The performed test scenarios show that SpaceWire with an adapted physical layer according to [11] is able to meet the requirements of both domains of robotics and big data. By using FPGAs as target platform for the SpaceWire implementation in robotic systems even the accessibility at arbitrary points in the network is possible. With different SpaceWire regions for control and recording in combination with routing switches following the crossbar switch design approach exclusive routes for both regions can be guaranteed. Due to this, the strict real time requirements for robotics can be fulfilled since the recording does not insert additional delay into the control communication path. With an adequate size of the data recorder's storage capability a balancing can be achieved in case of temporarily occupied resources on the recording host and loss of packets can be excluded. By integration of this approach into our SpaceWire eight-port switch, which have also been utilized to connect several robotic systems a fast and easy way for data recording on demand is created.

VI. OUTLOOK

Future work will consider the time aspect of the recorded packets. Since data transfers in our systems reference to a global time representation the process of recording packets can also reference to this global time. This enables, on the one hand, ordering of packets and on the other the determination of the transmission time of a specific packet through the network. An additional useful expansion for the data recording is also a matching unit, which enables the filtering of the recording packets according to the destination address. Another aspect which has to be considered in the future is the format of the saved raw packets. In this first implementation, a simple file approach is used (see section III). This may not be the most efficient way especially for more application specific data recording such as machine learning.

REFERENCES

- Juan C. Herrera, Daniel B. Work, Ryan Herring, Xuegang (Jeff) Ban, Quinn Jacobson, and Alexandre M. Bayen. Evaluation of traffic data obtained via gps-enabled mobile phones: The mobile century field experiment. *Transportation Research Part C: Emerging Technologies*, 18(4):568 – 583, 2010.
- [2] Arvind Thiagarajan, Lenin Ravindranath, Katrina LaCurts, Samuel Madden, Hari Balakrishnan, Sivan Toledo, and Jakob Eriksson. Vtrack: Accurate, energy-aware road traffic delay estimation using mobile phones. In Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems, SenSys '09, pages 85–98, New York, NY, USA, 2009. ACM.
- [3] Stefan Jörg, Mathias Nickl, Alexander Nothhelfer, Thomas Bahls, and Gerd Hirzinger. The computing and communication architecture of the dlr hand arm system. In 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 1055–1062. IEEE, 2011.
- [4] S. DiMaio and C. Hasser. The da vinci research interface. The MIDAS Journal - Systems and Architectures for Computer Assisted Interventions (MICCAI 2008 Workshop), July 2008.
- [5] P. Kazanzides, Z. Chen, A. Deguet, G. S. Fischer, R. H. Taylor, and S. P. DiMaio. An open-source research kit for the da vinci[®] surgical system. In 2014 IEEE International Conference on Robotics and Automation (ICRA), pages 6434–6439, May 2014.
- [6] M. Görner, T. Wimböck, A. Baumann, M. Fuchs, T. Bahls, M. Grebenstein, C. Borst, J. Butterfass, and G. Hirzinger. The dlr-crawler: A testbed for actively compliant hexapod walking based on the fingers of dlr-hand ii. In 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 1525–1531, Sept 2008.
- [7] U. Hagn, M. Nickl, S. Jörg, G. Passig, T. Bahls, A. Nothhelfer, F. Hacker, L. Le-Tien, A. Albu-Schäffer, R. Konietschke, M. Grebenstein, R. Warpup, R. Haslinger, M. Frommberger, and G. Hirzinger. The dlr miro: A versatile lightweight robot for surgical applications. *Industrial Robot: An International Journal*, 35(4):324–336, August 2008.
- [8] S. Thielmann, U. Seibold, R. Haslinger, G. Passig, T. Bahls, S. Jörg, M. Nickl, A. Nothhelfer, U. Hagn, and G. Hirzinger. Mica - a new generation of versatile instruments in robotic surgery. In 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, October 2010.
- [9] Mathias Nickl, Stefan Jörg, Thomas Bahls, Alexander Nothhelfer, and Stefan Strasser. Spacewire, a backbone for humanoid robotic systems. In *Proceedings of the 4th International SpaceWire Conference*, pages 356–359. Space Technology Centre, University of Dundee, 2011.
- [10] Secretariat ECSS. Spacewire-links, nodes, routers and networks. Technical report, ECSS-E-ST-50-12C, Noordwijk, The Netherlands, July, 2008.
- [11] Mathias Nickl, Stefan Jörg, and Thomas Bahls. Towards high-speed spacewire links. In *Proceedings of the 5th International SpaceWire Conference*, pages 263–266, 2013.
- [12] Stefan Jörg, Thomas Bahls, Mathias Nickl, and Stefan Strasser. Spacewire-hs host adapter-an fpga based pci express device for versatile high-speed channels. In *Proceedings of the 5th International SpaceWire Conference 2013*, pages 244–247, 2013.
- [13] Ulrich Hagn, Rainer Konietschke, Andreas Tobergte, Mathias Nickl, Stefan Jörg, Bernhard Kübler, Georg Passig, Martin Gröger, Florian Fröhlich, Ulrich Seibold, et al. Dlr mirosurge: a versatile system for research in endoscopic telesurgery. *International journal of computer* assisted radiology and surgery, 5(2):183–193, 2010.

Approach to Increase SpaceFiber Link Bandwidth Usage for Streaming Data Transfer

SpaceFibre, Short Paper

Ilya Korobkov, Irina Lavrovskaya Saint-Petersburg State University of Aerospace Instrumentation Saint Petersburg, Russia {ilya.korobkov, irina.lavrovskaya}@guap.ru

Abstract — High-rate SpaceFibre networks could be used to deliver intensive streaming data in modern spacecraft systems. Onboard sensor data streams or motion video traffic should be transmitted with small delays and jitter. The efficiency of data transfer over the SpaceFibre network could depend on the applied Transport protocol at nodes and traffic management at switches over SpaceFibre.

The paper is concerned with the application of ESDP (STP-2) Transport protocol for streaming. The paper also presents the traffic management over SpaceFibre. The traffic management is based on the traffic control via configuration parameters of SpaceFibre Management Information Base. To evaluate its benefits the simplified mathematical model of the streaming data transfer through Virtual Channels is proposed. The mathematical model is based on Markovian chain. The state graph of the model is shown. The probability of data transfer is made for steady-state conditions. The probability will specify how the quantity of transmitted data will be changed by the traffic control.

Index Terms — Spacecraft, Onboard Networks, Streaming, SpaceFibre, ESDP, STP-2, Traffic Control, Markovian chain.

I. INTRODUCTION

Every data stream in onboard spacecraft network is required a particular Quality of Service (QoS) [1-3]. Mission critical data, science data, live streaming interactive video specify special requirements for data delivery speed, latency and jitter [4]. In general, onboard spacecraft networks consist of nodes (data sources/targets) and switches. To perform data streaming transfer it is needed to define which network standard and transport layer protocol will be used for streaming.

II. STREAMING OVER ONBOARD NETWORKS

A. Network Standard For Streaming

Usually the use of various types of QoS adjusted to the traffic characteristics and the requirements of mission specific applications is needed. As demonstrated in [2, 4-9] the SpaceFibre spacecraft standard could be used for streaming. The SpaceFibre QoS mechanisms (Priority, Bandwidth Reservation and Schedule) meet

the QoS requirements for streaming data [2, 3, 6].

B. Transport Layer Protocol for Streaming

Energia Streaming Data Protocol (ESDP) is proposed as streaming oriented transport layer protocol for the SpaceFibre networks. "ESDP" is a new name of STP-2 protocol. It was designed to most closely meet the streaming protocol requirements of aerospace industry: compactness and simplicity of implementation, periodical packet issue and fixed packet size, packets with short headers, etc. The review and comparison of ESDP (under acronym "STP-2") with other protocols has already been done and presented in [2].

C. Traffic Management in SpaceFibre Networks

The highest priority for the onboard spacecraft networks is the ability to work correctly with failed ports, links and switches. When one port of a switch is failed, traffic could be redirected to another functional port. Which makes it necessary to be able to manage traffic over SpaceFibre networks. For this purpose, Adaptive Data Streaming Service (ADSS) [3] is proposed. ADSS performs adaptive control of QoS in the onboard network. ADSS provides two functions: network monitoring and QoS control. ESDP protocol and SpaceFibre standard are considered for ADSS functions.

1) Monitoring: It is used to evaluate the real network status: number of incorrect received packets, workload status of switch buffers, etc. This function could be implemented by ESDP and SpaceFibre.

ESDP protocol provides end-to-end feedback on the QoS in streaming data distribution by periodically (or by request) sending statistics information. For this purpose Status service packets are issued by the receiver/transmitter node.

A SpaceFibre interface includes a number of input and output virtual channels (VCs). When a data packet is placed in a SpaceFibre output VC it is transferred over the SpaceFibre link and placed in the same numbered input VC at the other end of the link. Moreover, SpaceFibre standard has the Management Information Base, which contains status parameters. When a VC is using less (under use

status) bandwidth than expected, it will be indicated in the status parameter [10].

2) *QoS Control:* The QoS control performs traffic control via configuration parameters of protocols/standards. Taking into account monitoring information and traffic requirements (rate, latency requirements, etc.) it could perform the following operations: increase or decrease priority; switch one QoS mechanism to the other; change time-slots table; switch connection-oriented data transfer to datagrams; etc [3].

In order to limit streaming flows ESPD implements Stop and Start service packets. The Stop packet stops data transmission. The Start packet starts (restarts) data transfer from the transmitter. ADSS could use these service packets for traffic management.

The Management Information Base of SpaceFibre includes QoS configuration parameters for VCs: priority level, time-slots, expected bandwidth, etc. These parameters could be used by ADSS for traffic management. It conforms to the SpaceFibre specification [10]. If the VC is using less bandwidth than expected, ADSS, for example, can dynamically increase priority level or change schedule table of VC to increase SpaceFibre link usage for streaming data transfer.

To evaluate benefits of ADSS – how much the bandwidth of VC will increase after applying ADSS – it is necessary to develop the mathematical approach to quantity calculation of transmitted streaming data via SpaceFibre VC. Also, this approach would make the process of the SpaceFibre network configuration (priorities, time-slots, etc.) much easier.

III. APPROACH TO QUANTITY CALCULATION OF TRANSMITTED STREAMING DATA VIA SPACEFIBRE OUTPUT VIRTUAL CHANNELS

A. Quantity Calculation of Transmitted Streaming Data via SpaceFibre Output Virtual Channels

Approach to quantity calculation of transmitted data via output VCs is divided into several steps which are described below in this subsection.

1) Schedule analyzing and subschedules defining: In this paper, a "sub-schedule" is a part of the schedule during which the values of the configuration parameters (that determine whether the VC is assigned to the time-slot, priority, expected bandwidth) are permanent, not changed. If such parts are available, then they form a sub-schedule for VCs. If there are no such parts, then all schedule is further considered as a common sub-schedule for all VCs.

Example: there are 2 VCs. The priorities and time-slots (schedule table) are presented in Table I. As a result of the current step, 3 sub-schedules will be defined: 0 - 32 time-slots; 33 - 53 time-slots; 54 - 64 time-slots.

2) Calculating of transmitted data: For each sub-schedule the average amount of transmitted data over the SpaceFibre output VCs is calculated. For this purpose the mathematical model of the streaming data transmission over output VCs for sub-schedule was developed. It is presented in the following subsections.

TABLE I. DEFINING SUB-SCHEDULES

TIME SLOTS	0		29	30	31	32	33	34	35		53	54		63
VC1				1					0			1	1	1
VC2	0	0	0		0	0	1	1	1	1	1	1	1	1
Sub-schedules		1				2					3			

3) Summation: The total value of the average amount of transmitted data over output VCs is defined as the sum of all volumes calculated for each sub-schedules.

B. Math Model of Streaming over Output Virtual Channels

The process of streaming data transmission over output VCs could be represented as a queuing system (system, QS). The QS receives incoming requests (arrivals, calls). They form *N* streams with intensities: λ_I , λ_2 , ..., λ_{N-I} , λ_N (Fig. 1). Each request is a data unit. The size of data unit does not exceed 256 bytes. The group of data units within a single stream forms a SpaceFibre packet. Each requests' stream corresponds to only one class of requests. Each requests class corresponds to only one VC. The total number of VCs is up to 256 VCs. In fact, only 32 VCs are used, the rest – reserved. Requests of different classes are buffered into different queues. All queues have the same limited capacity. The queue size varies from 1 to K requests. Each queue looks like a FIFO buffer of VC.



Fig. 1. Queue system of Output Virtual Channels SpaceFibre with MAC

The request service is the process of transmitting a data unit read from queue and then encapsulated into a data frame. The duration of data frame transmission does not depend on the request class. It is the same for all request classes. Requests are serviced with μ intensity.

The following assumptions are introduced: Data Link Layer of the SpaceFibre standard in the proposed mathematical model is considered; only Output Virtual Channels and Medium Access Controller mechanisms of Data Link Layer are considered; VCs are always notified that there is a room for a data frame at the other end of the link; it is not considered the case when Bandwidth Credit reaches values: \pm Bandwidth Credit Limit or Minimum Bandwidth Credit Threshold; the process of writing/reading data into/from VC buffer and generating data frame is not considered, delays are not considered; incoming requests' streams are determined by Poisson process; service times have an exponential distribution.

The intensity of incoming stream λ_i is calculated by the following formula:

$$\lambda_i = \frac{Size}{256 \cdot Period},\tag{1}$$

where λ_i – the intensity of incoming requests' stream; Size – size of
SpaceFibre packet, byte; 256 is the constant of max.size of a data frame; *Period* – the average period (cycle) of receipt of SpaceFibre packet.

The service time is defined by the following formula:

$$\mu = \frac{1}{t}, \quad t = \frac{S}{V}, \tag{2}$$

where μ – the service intensity; t – the duration of the service time, μ s; S – the size of SpaceFibre data frame, bits; V – speed of SpaceFibre link, bit/ μ s.

There are priorities between the different request classes. It means that a request with higher priority is served before a request with lower priority. Once the request service is started, it cannot be disrupted until the whole service requirement is completed. The QS described above is a Markov random process with continuous time (continuous Markov chain). The states of the Markov chain could be represented as:

$$X = I \cup S, \tag{3}$$

where the disjoint sets I and S describe the states of the system:

- *I* system is empty, all queues are empty;
- *32.* The state $S = \{S_1, S_2, ..., S_{N-1}, S_N\}$ is a finite set of system busy

states:

- S_I it services a request of the 1st request class;
- •••
- S_N it services a request of the Nth request class.

Each state S_i is an enlarged state, because each busy state S_i is divided into own set of queue states: $V = \{V_0, V_1, ..., V_L\}$ $V \subseteq S_i$ For easy reading all states of V_k ($0 \le k \le L$), they do not contain phrase "it services a request of *i* request class", but a reader have to bear it in mind:

- $V_0: 0/0/.../0/0$ all queues are empty;
- *V₁*: 1/0/.../0/0 one request is waiting for the service in the queue of the 1st request class, other queues are empty;
- •••
- V_K: K/0/.../0/0 queue of the 1st request class is full, other queues are empty;
- $V_{K+K+...+K}$: K/K/.../K/0 queues of the 1st, 2nd, ..., N-2th, N-1th request classes are full, another queue is empty;
- •••
- $V_{L=K+K+...+K+K}$: K/K/.../K/K all queues are full.

The state diagram of the whole system with queues' details is presented in Fig. 2.

The distinctive feature of the proposed mathematical model is the set of probabilities $R_{i_0,i_1,...,i_{N-1},i_N,i_{N+1}}$ ($i_a = 0,1,...,N$; a = 1,2,...,N+1). It is specified for each S_i state with respect to V_k queue state. The following index notation is used: i_0 – it specifies a request class's number of served request: $1 \le i_0 \le N$; $i_1,...,i_{N-1},i_N$ – indexes that

define one particular state of the queue states of V_k : $i_b = 0, 1, 2, ..., K; b = 1, 2, ..., N; i_{N+1}$ – it specifies index of a probability to make the transition of other states: $1 \le i_{N+1} \le N$.



Fig. 2. The state diagram of the whole system

The probabilities $R_{i_0,...,i_{N+1}}$ allow to make the transition to service of a request from the queue of i_b request class. Such kind of transition will move the system to one state of queue states V_k ($0 \le k \le L$) within the current enlarged state S_j ($j = i_b$). It also permits to move the system from the current enlarged state S_j to S_{i_b} ($j \ne i_b$). As a result, the system will be transferred to one state of queue state V_k for the enlarged state S_{i_b} . The function F is introduced to determine the probabilities $R_{i_0,...,i_{N+1}}$:

$$R_{i_0,\dots,i_{N-1}} = F(M, E, C, V), \qquad (4)$$

where M – priority levels of all VCs: $M = \{M_1, M_2, ..., M_N\}$ $0 \le M_i \le 16$; E – percentage of overall link bandwidth that VC is expected to use: $E = \{E_1, E_2, ..., E_N\} \sum_{i=1}^{N} E_i \le 0.9$; C – maximum

amount of link bandwidth that VC is allowed to accumulate; V – numbers of queues that contain requests for service.

Using the state diagram shown in Fig. 2,**Error! Reference** source not found. the system of differential Kolmogorov's equations could be obtained. It could be transformed to the system of linear equations (SLE). It is necessary to point out that the following situation is considered [11]: there are the steady-state conditions; there are steady-state probabilities of the system states. The solution of SLE is the steady-state probabilities $p_{i_0,i_1,...,i_{N-1},i_N}$ (index notation like above mentioned). These probabilities mean the average relative time of request service time. This time determines the whole duration of request transmitting of i_0 request class. The SLE equations are presented:

$$\begin{cases} \sum_{i=1}^{N} \lambda_{i} p_{0,0,\dots,0,0} = \mu \sum_{i=1}^{N} p_{1,0,\dots,0,0}, \\ \left(\sum_{i=1}^{N} \lambda_{i} + \mu \right) p_{1,0,\dots,0,0} = \lambda_{1} p_{0,0,\dots,0,0} + \mu \sum_{i=1}^{N} p_{i,1,\dots,0,0}, \\ \left(\sum_{i=1}^{N} \lambda_{i} + \mu \right) p_{1,q,\dots,0,0} = \lambda_{1} p_{1,q-1,\dots,0,0} + \mu \sum_{i=1}^{N} p_{i,q+1,\dots,0,0}, q = \overline{1, K-1}, \\ \left(\sum_{i=2}^{N} \lambda_{i} + \mu \right) p_{1,K,\dots,0,0} = \lambda_{1} p_{1,K-1,\dots,0,0}, \\ \dots \\ \left(\sum_{i=N-1}^{N} \lambda_{i} + \mu \sum_{j=1}^{N-1} R_{1,K,\dots,q,0,j} \right) p_{1,K,\dots,q,0} = \lambda_{1} p_{1,K-1,\dots,q,0} + \dots + \lambda_{N-1} p_{1,K,\dots,q-1,0}, q = \overline{1, K-1}, \\ \left(\lambda_{N} + \mu \sum_{j=1}^{N} R_{1,K,\dots,K,0,j} \right) p_{1,K,\dots,K,0} = \lambda_{1} p_{1,K-1,\dots,K,0} + \dots + \lambda_{N-1} p_{1,K,\dots,K-1,0}, \\ \dots \\ \left(\lambda_{N} + \mu \sum_{j=1}^{N} R_{1,K,\dots,K,0,j} \right) p_{1,K,\dots,K,q} = \lambda_{1} p_{1,K-1,\dots,K,q} + \dots + \lambda_{N} p_{1,K,\dots,K-1,0}, \\ \left(\mu \sum_{j=1}^{N} R_{1,K,\dots,K,K,j} \right) p_{1,K,\dots,K,K} = \lambda_{1} p_{1,K-1,\dots,K,K} + \dots + \lambda_{N} p_{1,K,\dots,K,K-1}, \\ \dots \\ \left(\mu \sum_{j=1}^{N} R_{N,K,\dots,K,K,j} \right) p_{N,K,\dots,K,K} = \lambda_{1} p_{N,K-1,\dots,K,K} + \dots + \lambda_{N} p_{N,K,\dots,K,K-1}, \\ \dots \\ \left(\mu \sum_{j=1}^{N} R_{N,K,\dots,K,K,j} \right) p_{N,K,\dots,K,K} = \lambda_{1} p_{N,K-1,\dots,K,K} + \dots + \lambda_{N} p_{N,K,\dots,K,K-1}, \\ \dots \\ \left(\mu \sum_{j=1}^{N} R_{N,K,\dots,K,K,j} \right) p_{N,K,\dots,K,K} = \lambda_{1} p_{N,K-1,\dots,K,K} + \dots + \lambda_{N} p_{N,K,\dots,K,K-1}, \\ \dots \\ \left(\mu \sum_{j=1}^{N} R_{1,0,\dots,0,j} + \sum_{j=0}^{K} p_{1,0,\dots,1,j} + \dots + \sum_{j=0}^{K} p_{N,K,\dots,K,j} = 1 \\ \sum_{j=0}^{N} p_{N,K,\dots,K,K,j} = 1 \\ \sum_{j=0}^{N} p_{N,K,\dots,K,K,j} = n$$

From knowledge of the average duration of the service time and size of request in bytes the average quantity of transmitted data of i_0 request class could be calculated by the following formula:

$$AQ_{i_0} = \frac{T_{i_0} \cdot Length_{i_0}}{T_{frame} \cdot Frames_{i_0}},$$
(6)

where AQ_{i0} – the average quantity of transmitted data of i_0 request class, bytes; T_{i0} – the average duration of the requests' service time of i_0 request class, ms; T_{frame} – the average time for sending a data frame, ms; *Length*_{i0} – the length of packet of i_0 request class, bytes; *Frames*_{i0} – the quantity of frames inside one packet of i_0 request class.

 AQ_{io} demonstrates the average amount of transmitted data over output VCs, because request classes are brought in line with VCs.

C. Example of Calculations

There are considered 2 VCs, sizes of VCs buffers are up to 1 request. Traffic parameters are demonstrated in Table II. Parameters of VCs are presented in Table III.

TABLE II. TRAFFICS PARAMETERS

Request classes	Size, byte	Quantity frames in packet	Average period of receipt, us
VC1	1024	4	20
VC2	260	1,015625	5000

TABLE III. VCS PARAMETERS

Priority level	Expected Bandwidth	0	•••	15	16	17		31	32	33	•••	46	47	•••	63
1	0,032768				1				1				1	1	1
0	0,0003328	0	0	0		0	0	0		0	0	0	0	0	0
Duration of 1 time-slot, us							2	60	,93	75					
SpaceFibre link, Gbit/sec								1	,25						
1	level 1 0 n of 1 tin ibre link	Ievel Bandwidth 1 0,032768 0 0,0003328 n of 1 time-slot, us ibre link, Gbit/sec	IntervelDependence 0 levelBandwidth1 $0,032768$ 0 $0,0003328$ 0 $0,0003328$ 0ibre link, Gbit/sec	IevelBandwidth01 $0,032768$ 0 $0,0003328$ 0 $0,0003328$ 0 0 n of 1 time-slot, usibre link, Gbit/sec	Iterel Bandwidth 0 15 1 0,032768 0 0 0 0 0,0003328 0 0 0 n of 1 time-slot, us ibre link, Gbit/sec 0 0	Iterest Bandwidth 0 15 16 1 0,032768 1<	Iterel Bandwidth 0 15 16 17 1 0,032768 1 1 0 </td <td>Iterel Bandwidth 0 15 16 17 1 0,032768 1</td> <td>Iterel Bandwidth 0 15 16 17 31 1 0,032768 1</td> <td>Iterel Bandwidth 0 15 16 17 31 32 1 0,032768 1 1 1 1 1 1 1 1 1 1 1 1 0 0,0003328 0</td> <td>Iterel Bandwidth 0 15 16 17 31 32 33 1 0,032768 1 1 1 1 1 1 0 <t< td=""><td>Initial Problem Description 0 15 16 17 31 32 33 1 0,032768 1</td></t<><td>Initial Problem Description Image: Constraint of the image: Constraintof the image: Cons</td><td>Initial Problem Description <thdescription< th=""></thdescription<></td><td>Initial Problem Description <thdescription< th=""></thdescription<></td></td>	Iterel Bandwidth 0 15 16 17 1 0,032768 1	Iterel Bandwidth 0 15 16 17 31 1 0,032768 1	Iterel Bandwidth 0 15 16 17 31 32 1 0,032768 1 1 1 1 1 1 1 1 1 1 1 1 0 0,0003328 0	Iterel Bandwidth 0 15 16 17 31 32 33 1 0,032768 1 1 1 1 1 1 0 <t< td=""><td>Initial Problem Description 0 15 16 17 31 32 33 1 0,032768 1</td></t<> <td>Initial Problem Description Image: Constraint of the image: Constraintof the image: Cons</td> <td>Initial Problem Description <thdescription< th=""></thdescription<></td> <td>Initial Problem Description <thdescription< th=""></thdescription<></td>	Initial Problem Description 0 15 16 17 31 32 33 1 0,032768 1	Initial Problem Description Image: Constraint of the image: Constraintof the image: Cons	Initial Problem Description Description <thdescription< th=""></thdescription<>	Initial Problem Description Description <thdescription< th=""></thdescription<>

There is only one sub-schedule: 0 - 63 time-slots. The intensities are: $\lambda_1 = 0,00020312$; $\lambda_2 = 0,2$; $\mu = 0,473485$. The state diagram is shown in Fig. 3.



Fig. 3. The state diagram of the system with 2 VCs and buffer for 1 request

The SLE equations are presented below:

$$\begin{cases} (\lambda_{1} + \lambda_{2}) p_{0,0,0} = \mu(p_{1,0,0} + p_{2,0,0}), \\ (\mu + \lambda_{1} + \lambda_{2}) p_{1,0,0} = \lambda_{1} p_{0,0,0} + \mu(p_{1,1,0} + p_{2,1,0}), \\ (\mu + \lambda_{2}) p_{1,1,0} = \lambda_{1} p_{1,0,0}, \\ (\mu + \lambda_{1}) p_{1,0,1} = \lambda_{2} p_{1,0,0} + \mu(p_{1,1,1} + p_{2,1,1}), \\ \mu p_{1,1,1} = \lambda_{2} p_{1,1,0} + \lambda_{1} p_{1,0,1}, \\ (\mu + \lambda_{1} + \lambda_{2}) p_{2,0,0} = \lambda_{2} p_{0,0,0} + \mu(p_{1,0,1} + p_{2,0,1}), \\ (\mu + \lambda_{1}) p_{2,0,1} = \lambda_{2} p_{2,0,0}, \\ (\mu + \lambda_{2}) p_{2,1,0} = \lambda_{1} p_{2,0,0}, \\ \mu p_{2,1,1} = \lambda_{1} p_{2,0,1} + \lambda_{2} p_{2,1,0}. \end{cases}$$

$$(6)$$

$$p_{0,0,0} + \sum_{j=0}^{K=1} p_{1,0,j} + \sum_{j=0}^{K=1} p_{1,1,j} + \sum_{j=0}^{K=1} p_{2,0,j} + \sum_{j=0}^{K=1} p_{2,1,j} = 1$$

The solution of the equations is shown in formula (7).

After substituting numbers, the steady-state probabilities of all system states are: $p_{0,0,0} = 0.624319541$; $p_{1,0,0} = 0.000244197$; $p_{1,1,0} = 0.000000074$; $p_{1,0,1} = 0.00018455$; $p_{1,1,1} = 0.00000011$; $p_{2,0,0} = 0.26373621$; $p_{2,0,1} = 0.111354404$; $p_{2,1,0} = 0.000079544$; $p_{2,1,1} = 0.00008137$.

If the running time of SpaceFibre switch is 500 ms, the average duration of the service time of VC1 requests (taking into account time-slots) will be 55.7 ms; the average service time of VC2 requests is 0.21 ms. The quantity of transmitted data over VCs: $AQ_{VC1} = 6.4$ MBytes; $AQ_{VC2} = 24.6$ KBytes. It could be converted to VCs throughputs: 11.8 KBytes/ms and 0.05 KBytes/ms.



IV. CONCLUSION

The paper gives a short overview of actual problem of streaming data transfer over onboard spacecraft networks. The SpaceFibre standard and ESDP transport protocol could be used for streaming. The concept of traffic management by means of SpaceFibre and ESDP mechanisms was proposed. The mathematical approach to quantity calculation of transmitted streaming data via SpaceFibre was developed to evaluate how much data will be transferred via SpaceFibre. The mathematical model of the streaming data transfer over SpaceFibre output virtual channels was also proposed. All these results are actual and significant for designers of spacecraft networks. It would make the process of evaluation of the SpaceFibre network characteristics and generating network configuration parameters (priorities, time-slots, etc.) much easier.

ACKNOWLEDGMENT

The research leading to these results has received funding from the Ministry of Education and Science of the Russian Federation within the project part of the federal assignment under contract N 8.4048.2017/4.6 from 31.05.2017.

REFERENCES

- I. Korobkov, "Adaptive Control of the Network Protocols' Mechanisms in the Onboard Networks", Proceedings of SUAI Scientific session, Saint-Petersburg State University of Aerospace Instrumentation, Saint-Petersburg, SUAI, 2015, pp. 73-82.
- [2] I. Korobkov, E. Suvorova, Y. Sheynin, V. Olenev "Streaming Services over

SpaceFibre Networks", Proceedings of 7th International SpaceWire Conference 2016 (ISC2016), Yokohama, Japan, 2016. pp. 151-158.

- [3] I. Korobkov "Adaptive Data Streaming Service for Onboard Spacecraft Networks", Proceedings of 17th Conference of Open Innovations Association Finnish-Russian University Cooperation in Telecommunications (FRUCT), P.G. Demidov Yaroslavl State University, Yaroslavl, Russia, 2015, pp. 291-298.
- [4] A. Khakhulin, I. Orlovsky, Y. Sheynin, I. Korobkov, V. Olenev, E. Suvorova, I.Lavrovskaya "SpaceFibre Based On-board Networks for Real-Time Video Data Streams", Proceedings of 7th International SpaceWire Conference 2016 (ISC2016), Yokohama, Japan, 2016, pp. 327-333.
- [5] N. Matveeva, E. Suvorova, Y. Sheynin, "QoS mechanisms in SpaceFibre and RapidIO Services over SpaceFibre Networks", Proceedings of 7th International SpaceWire Conference 2016 (ISC2016), Yokohama, Japan, 2016, pp. 305-312.
- [6] I. Korobkov "A Queuing System for the SpaceFibre Standard", Proceedings of 21th Conference of Open Innovations Association Finnish-Russian University Cooperation in Telecommunications (FRUCT), Jyvaskyla, Finland, 2018, in press.
- [7] A. Shamshin, I. Korobkov, N. Matveeva "Evaluations of the SpaceFibre QoS Mechanisms for Streaming", Proceedings of SUAI Scientific session, Saint-Petersburg State University of Aerospace Instrumentation, Saint-Petersburg, SUAI, 2015, pp. 178-191.
- [8] E. Suvorova, N. Matveeva, I. Korobkov, A. Shamshin, Y. Sheynin, "Virtual Prototyping in SpaceFibre System-on-Chip Design", Design and Verification Conference and Exhibition (DvCon) Europe 2015, Munich, Germany, 2015.
- [9] I. Korobkov, E. Suvorova, N. Matveeva, Y. Sheynin, "Virtual Prototyping for Spacefibre Standard Research", Izvestia of Samara Scientific Center of the Russian Academy of Sciences, volume 18, No.1(2), Samara, Russia, 2016, pp. 390-396.
- [10] S. Parkes, A. Ferrer, A. Gonzalez, C. McClements, SpaceFibre Specification. Draft K1, 2018, 246 p.
- [11] E. Ventsel, Operations Research. Moscow: Soviet Radio, 1972, 552 p.

Wideband DVB-S2X for EOS Downlink on SpaceFibre-Interconnected Dual RC64

Peleg Aviely, Tsvika Israeli, Gilad Danin, Moshe Goren, Fredy Lange, and Ran Ginosar

Ramon Chips Ltd. Yoqneam Illit, Israel peleg@ramon-chips.com

Abstract- RC64 is a rad-hard manycore DSP combining 64 VLIW/SIMD DSP/CPU cores, 4MB lock-free EDAC-protected shared memory, a hardware scheduler, and a task-based programming model. The hardware scheduler enables fast scheduling and allocation of fine grain tasks to all cores. RC64 achieves 50 16-bit GMACS and 25 single precision GFLOPS, 120Gbps SpaceFibre connectivity using 12 links (each with up to 5Gbps full duplex data rate), 38Gbps LVDS exchange with ADC/DAC, and over 16Gbps data rate to EDAC-protected external DDR3 memory. The demonstration platform for RC64 is based on a VPX board, a customized chassis, and high-end interfaces of RC64 to communication links and storage components. Software applications and libraries for the platform provide preliminary performance and power results. This article includes details of the demonstration platform and benchmark results for a DVB-S2X wide band transmitter.

Index Terms-DVB-S2X, EOS, VPX, RC64, SpaceFibre

I. VPX64TX DEMONSTRATION SYSTEM

Fig. 1 shows the VPX64tx board. Fig. 2 and Table I describe the board architecture.

Fig. 3 describes the integration of two VPX64tx boards with a VPX3UVP6 backplane, fitting into a 3U VPX C-Frame chassis. For example, Slots #2 through #6 contain two boards on Slots #4 and #5, where Slot #5 connects through the front panel using an RC64 SpaceWire Host #0 interface to a Brick Mk3. A USB3 interface connects the brick to the controlling PC. The second board connects through the SpaceWire host ring in the backplane. The software loaded to the board in Slot #5 routes SpaceWire packets to the board in Slot #4 to/from the PC. The optional debugger box connects through the backplane connector to the RC64 to support debug sessions.



Fig. 1. VPX64tx board

TABLE I. ITEMS SHOWN IN FIG. 2

Item	Description
DAC	The RC64 processor connects to two digital-to-analog converter (DAC) components (for Q and I output), each with
	24 Low Voltage Differential Signaling (LVDS) data pairs
	(DIN) and one data valid (IDC) input pair, using a dedicated
	clock generator to control DAC frequency and phase
	alignment.
DDR3	The volatile memory array is for general-purpose computing.
Flash	The non-volatile flash memory is for boot and data that must
	be retained.
SpW	Two SpaceWire host control interfaces (SpW Host #0, #1)
host	with configurable redirection of one interface (SpW Host #0)
	to the front panel.
SpW	Four SpaceWire data interfaces for backplane
data	interconnections.
SpFi	All 12 SpaceFibre interfaces connect to the backplane.
	Optional: one of the interfaces (SpFi #0A) connects to the
	front panel.
JTAG	The debug interface connects to the backplane. Optional: on-
	board connection. The on-board switch enables debug mode.



Fig. 3. VPX64tx demonstration system

II. DVB-S2X TRANSMITTER

The DVB-S2 transmitter software (see [1]) has a pipeline design consisting of the following stages:

- Data packet input
- Bose–Chaudhuri–Hocquenghem (BCH) frame encoding
- Low-density parity-check (LDPC) frame encoding
- Symbol generation
- Shaping and interpolation filter
- DAC output

The pipeline utilizes a double-buffer mechanism, which resides on the shared memory to pass data between the stages. The final DAC output buffer size determines the maximum number of samples that are created in a single iteration, according to the chosen Physical Layer Signaling (PLS) and the Samples per Symbol (SpS) ratio. The samples are complex numbers, 16bit-I, 16bit-Q. Using an SpS ratio of 3 and a final buffer size of 820Kbytes, we arrive at the values per iteration shown in Table II.

Table III, Fig. IV, and Fig. V describe the six tasks, the task dependency graph, and the execution schedule on 64 cores of RC64.

TABLE II. PLS PARAMETERS

PLS	Name	Frames	Total Symbols	Total Samples	Iteration Cycles
43	QPSK 8/9	8	66,960	200,880	51,300
67	8PSK 8/9	12	67,176	201,528	51,300
111	32APSK 8/9	20	68,040	204,120	51,300

TABLE III.	DVB-S2 TASK DESCRIPTIONS
------------	---------------------------------

Task Name	Task Type	Task Priori	Pipe Stage	Task Function	Task Parameters	Performance (PLS=111)	Instances Per Iteration
		ty					
DMA In	High priority task, starts at SpFi end-of- buffer receive	1	1	Collect frames payload in buffers	N/A	Throughput limited by modem	Asynchronous task, input data rate dependent
LDPC encode	Regular task activates the LDPC accelerator	2	3	Manage LDPC encoder DMA	PLS number defines LDPC code rate for hardware LDPC accelerator	~1300 cycles per frame	1
BCH frame encoding	Duplicable task, processes two frames per task instance	3	2	Input raw bits to BCH frame including scrambling and BCH encoding; prepare bits for LDPC encoder	PLS number defines BCH polynomial and number of bits per frame	Cycles per bit: 10cycles/8bits for two frames. ~40,000 cycles per two frames	10
Symbols to samples	Duplicable task processes range of output samples	4	5	Symbols to samples including Farrow re- sampling at 3:1 samples to symbols rate	Shaping filter taps value, samples per symbol ratio	~45,000 cycles per instance	51
Frame bits to symbols	Duplicable task, processes N frame slots per task instance	5	4	Frame bits to symbols including bit interleaving, pilot and header insertion, symbol mapping, and symbol scrambling	PLS number defines header symbols, interleaving, constellation mapping, and symbols number	~6,300 cycles per instance	72
DMA Out	High priority task, starts at end of buffer transmit	6	6	Assign ready-to-send samples from double buffer in shared memory	Number of samples in buffer	54,000 cycles for 750Msamples/sec System clock = 200MHz	Asynchronous task, output data rate dependent



Fig. 4. Task dependency graph

I. TRANSMITTER PERFORMANCE

Fig. 5 is the recorded schedule of tasks during two iterations time for transmitting a full buffer of the selected PLS (see Table II). 750Msamples/sec of I/Q samples at 3SpS represents 250MSymbols/sec and translates to a bandwidth of roughly 500MHz.

PLS 111 with a 32APSK-8/9 short frame configuration transmits 20 frames per 54,400 cycles, each containing 14,232 bits at a rate of 14,232 bits x 20 frames / 54,400 cycles x 200M cycles/sec = 1.046Gbps.

PLS 43 with a QPSK-8/9 short frame configuration transmits 8 frames per 54,400 cycles at a rate of 425 Mbps.

The average measured RC64 power for PLS 111 execution is roughly 10W.

Fig. 6 describes the data flow of the transmitter with a single RC64 processor on the VPX64tx board. Two DACs generate a baseband I/Q analog signal, connected to an up converter or measurement equipment. Comparing the measured spectrum and recorded output to the expected spectrum and a MATLAB simulation achieves accurate results.



Fig. 5. Task scheduling





III. TRANSMITTER SCALING TO 1GHZ BANDWIDTH

The VPX64tx board clock generator supports a programmable clock rate, achieving a range of sample rates. The compute rate is defined by the longest task execution time per iteration. Scale-down implementations can use a slower system clock for slower sample clock rates to minimize dynamic power. Scale-up implementations can distribute the

computation over multiple RC64 processors on different boards, using SpaceFibre links to communicate intermediate data between pipe stages.

Fig. 7 describes the data flow of the transmitter with two RC64 processors on two VPX64tx boards, integrated with SpaceFibre interfaces. Table IV shows the computation split.



Fig. 7. Two-board transmitter

Fig. 8.

Task	Board A	Board B	SpaceFibre
			Communication
DMA in	Input buffer and DMA control	N/A	Up to 3Gbps, packet per frame
Split even and odd frames	Buffer and SpaceFibre DMA control: send even frames to board B	Buffer and SpaceFibre DMA control: receive even frames from board A	Up to 1.5Gbps, packet per frame
Bits to BCH frame	Process even frames	Process odd frames	N/A
LDPC encoder	Process even frames	Process odd frames	N/A
Frame bits to symbols	Process even frames	Process odd frames	N/A
Split I/Q symbols	Buffer and SpaceFibre DMA control: send	Buffer and SpaceFibre DMA control: send	Board A to B: 4Gbps
	odd Q symbols, receive even I symbols	even I symbols, receive odd Q symbols	Board B to A: 4Gbps
Symbols to samples	I symbols to samples	Q symbols to samples	N/A
DMA out	Output buffer and DMA control to DAC-I (only one of the DACs is active)	Output buffer and DMA control to DAC- Q (only one of the DACs is active)	N/A

TABLE IV. Two VPX64TX boards transmitter task split

The clock generators on each board allow synchronizing DAC-I and DAC-Q sampling. A reference input clock sets the same sample frequency, and the two clock generators start clock generation with a common sync signal driving both boards from a common source.

IV. CONCLUSIONS

RC64 is a high performance, manycore DSP suitable for use in space. It enables a wide range of applications, while providing high level protection from radiation effects. The wide band DVB-S2X transmitter application demonstrates diverse computational tasks with two boards. The 250Msymbol/sec throughput of a single VPX64tx board can scale up to 500Msymbols/sec on two boards connected with SpaceFibre. The demonstration system shows a 1GHz output baseband signal (1.35GHz including 1.35 roll-off factor). The maximum input data rate is 3Gbps, and the maximum power is 20W for RC64 and 6W for DACs and other peripherals.

REFERENCES

 P. Aviely, O. Radovsky, and R. Ginosar, "DVB-S2 software defined radio modem on the RC64 manycore DSP," in proceedings of International Astronautical Conference, pp. IAC-15-B2.6.1, Jerusalem, Israel, October 2015.

Implementation of SpaceWire Optical Fiber Link for High-speed and Long-distance Data Acquisition

Poster Session, Short Paper

Yi Xiaosu, Zeng Huasong, Zhang Chunxi School of Instrumentation Science and Opto-electrionics Engineering Beihang University Beijing, 100191,China. yixiaosu@buaa.edu.cn, zenghuasong@buaa.edu.cn, zhangchunxi@buaa.edu.cn

Abstract—SpaceWire is a high-speed data link designed specifically for spacecraft onboard data communication. It uses electric cable, supporting 200Mbit/s transmission rate, but the full-rate transmission distance is limited within 10m, which limits the future application and development. The implementation of SpaceWire Optical fiber data link, using optical fiber instead of electric cable, can significantly increase data transmission rate and distance of SpaceWire.

This paper studies the mechanism of implementing optical fiber as transmission medium based on SapceWire technology on physical and protocol layer. On physical layer, the optical transceiver, SerDes and FPGA techniques are integrated, providing a high-speed data transmission link. On protocol layer, an optical link interface controller VHDL IP core is designed based on SpaceWire protocol, with handshake mechanism modified to ensure that SerDes has enough time to lock the input signals. And the signal coding is changed from Data-Strobe coding to 8b/10b coding, to meet the requirement of very high speed data transmission and SerDes technology. The challenge of using new coding technique is also discussed. Then the paper analyzes the flow control mechanism of SpaceWire when transmission medium is optical fiber, builds a model to describe the relationship between FCT tokens, data transmission rate and optical fiber length, provides a modified technical way to implement the SpaceWire flow control mechanism, which can increase the full-rate transmission distance.

To test the SpaceWire optical fiber communication system implemented in this paper, a software is developed. The software controls SpaceWire packet transmission and monitors link status. Test result shows the implemented SpaceWire optical fiber link provides a 1.92Gbit/s data link, can achieve 1.5Gbit/s effective full-rate data transmission with 900 meter fiber length. The test shows the effectiveness of the research.

Index Terms—SpaceWire, Fiber communication, Transmission distance, Latency, Flow control

I. INTRODUCTION

SpaceWire^[1] is a data communication protocol designed specifically for spacecraft onboard network. It connects together sensors, memory, processing units and down link telemetry sub-system, provides high-speed (2 to 200 Mbits/s), bi-directional, full-duplex data links for data acquisition and transmission^[2]. Recently, SpaceWire link with higher speed is developing, such as GigaSpaceWire^[3], to meet the requirement of increasing transmission speed of data transmitted among components in the space system. However, the higher link speed will intensify the DC level drift, so galvanic isolation is needed between systems ^[4]. Fiber communication has the features of high bandwidth, low crosstalk, low attenuation, and also provides optoelectronic isolation, which is befitting for high-speed and long-distance data transmission. In fact, the advanced SpaceWire protocol with optical fiber has been developed, like SpaceFibre^[5].

In this paper, a high-speed SpaceWire fiber optic data link is implemented, with physical level, signal level and character level modified, which provides Gbits/s link rate. But in the application of long-distance data transmission, the wave propagation delay ^[6] will lead to the failure of SpaceWire's exchange of silence handshake process. This delay will also have the effect on the flow control in SpaceWire network, resulting in the effective data rate (Bandwidth Utilization) decreased. To solve these problems, the paper analyzes the signal latency in the relatively long optical fiber link and builds a distance-rate model. The possible ways to handle the problems when using new coding techniques are discussed.

II. STRUCTURE AND FUNCTION

The SpaceWire optical fiber communication system implemented in this paper is illustrated in Fig.1.

An Optical SpaceWire node mainly consists of 1x USB port, 1x CY7C68013 USB Microcontroller, 1x FPGA chip, 2x TLK2711 SerDes ICs and 2x SFP optical transceivers. The local node is connected with a computer through USB 2.0, and controlled by software. The two optical transceivers can be connected with remote node for data transmission, or they can be connected with each other for loop test.

SpaceWire Codec can be implemented on FPGA ^[7], and an Optical SpaceWire Codec IP core is designed in this paper. There's also a control logic implemented on the FPGA, which glues the USB interface and the Optical SpaceWire Codec, processes the commands received from the computer software and gives responses, controls the SpaceWire Codec according

to the commands. The Optical SpaceWire Codec provides a link interface to the SerDes for data exchange, and the SerDes drives the optical transceiver for high-speed serial data transmission. Thus, a high-speed SpaceWire optical fiber data link is established.



Fig. 1. SpaceWire optical fiber communication system.

The computer software provides data transmit/receive panel and a monitor panel. In the data transmit/receive mode, all data bytes which come from SpaceWire network pass through the USB link to the software and are displayed in textboxes, so the data rate is limited by the bandwidth of the USB link. In the monitor mode, data bytes received from SpaceWire network are counted by a statistic module in the FPGA, and then they are discarded without any processing. The software calculates the Optical SpaceWire data rate by reading the statistic registers every second, so the data rate of SpaceWire fiber link can be tested without the bandwidth limitation of USB link.

III. DESIGN OF OPTICAL SPACEWIRE CODEC

The structure of Optical SpaceWire Codec is shown in Fig.2.



Fig. 2. Optical SpaceWire Codec.

The Codec VHDL IP Core is implemented on the FPGA, it provides user interface and link interface. User interface includes link control interface, data interface, Time-Code interface and statistic interface. Link interface now provides data and control signal to SerDes and optical transceiver^[8], instead of the original SpaceWire Data-Strobe signal.

Although the SerDes recovery clock frequency equals to the local transmit clock frequency theoretically, there exists skew and jitter of recovery clock. Therefore, we consider that all the components in this Codec except the Receiver module are in the local clock domain, and signals from the Receiver should be synchronized to the local clock domain. So a faster clock drives synchronizers to sample the signals on the Receiver side^[9], to avoid meta-stability and data loss^[10].

Because of the change in physical level (using fiber optics instead of electric cable), using the original Data-Strobe coding might cause some problems. The first one is, although Data-Strobe coding can tolerant up to 0.5bit skew between data and strobe signal, under 200Mbps condition, the permitted skew time is 2.5ns. But if the link speed increases to 2Gbps, 2.5ns skew time equals to 5bits and it will certainly lead to a link failure. That means the skew have to be limited within 0.25ns if the Data-Strobe coding is used in the 2Gbps high-speed link, it is very hard to achieve. The second one is, data and strobe actually is a pair of physical independent signals, they always occupy two physical medium link, and it is not worth to couple them into one fiber using technologies like WDM. So implementing high-speed fiber optics data link with Data-Strobe coding is no longer applicable.

Therefore the signal level uses 8b/10b coding which is widely used in high-speed serial data link^[11] instead of Data-Strobe coding. In 8b/10b coding, the clock is embedded into data stream, so it is much easier to reduce the skew between data and clock signals. The character level remains unchanged, but the control characters and control codes have to be mapped into 8b/10b K-code. The mapping is shown in Fig.3.

SpaceWire Character		8b/10b Code
FCT	\rightarrow	K28.1
EOP	\rightarrow	K28.2
EEP	\rightarrow	K28.3
ESC	\rightarrow	K28.4
NULL	\rightarrow	K28.5
Silence	\rightarrow	K28.6

Fig. 3. Code Mapping.

In 8b/10b coding, SerDes receive circuit detects Comma(K28.1, K28.5, K28.7) for byte alignment^[12], it requires at least one Comma presented to receiver side before the normal transmission to make the receive data aligned. Further, during the transmission, there should be Commas transmitted regularly to keep the data aligned without error. So control character NULLs which are transmitted during the link initialization is mapped into K28.5 and the FCTs transmitted most frequently during normal data transmission in SpaceWire link are mapped into K28.1. In addition, it is normally for transmitter turning off the light to indicate silence, but the silence is just keep 19.2us in the SpaceWire state machine, it is impossible for most of the optical transceivers to turn on and off the light within such a short time. It is necessary to add a K-code to give transmitter the ability to indicate silence state with

the light source powered on. Therefore, K28.6 is involved to indicate the silence state, so the link resets and starts exchange of silence process both when K28.6 presents or LOS is detected and last for 850ns on the receiver side.

The data width of the Codec can be set to 8bits or 16bits to meet different system and data processing requirement. In 8bits mode, only the LSB of TLK2711 is used, the MSB is unused and can be connected to a random or fixed data byte. And in 16bits mode both LSB and MSB are used for data characters, but for controls characters, K-code should be always presented to the LSB, with the MSB left unused.

IV. DATA RATE AND DISTANCE ANALYSIS

The SpaceWire Optical link works well with short fiber optics. But in long distance application, the data latency caused by wave propagation time in long fiber optics brings impacts to the data link, the handshake process and flow control mechanism is affected especially.

A. Handshake Mechanism

During a handshake attempt cycle, the SpaceWire Codec keeps silent for $T_{Silence}=6.4us+12.8us$, and keeps sending NULLs for $T_{NULL}=12.8us$. In long distance application, due to the propagation time, it can be considered that continuous silence and NULLs have a length of $T_{Silence}$ and T_{NULL} in the transmission link. And $L_{NULL}=(c/n)\cdot T_{NULL}$, $L_{Silence}=(c/n)\cdot T_{Silence}$, in where *c* is the speed of light in vacuum, *n* is the refractive index of the fiber optics.



Fig.4. shows the NULLs transmitted in long fiber optics during a handshake attempt cycle, considering propagation time. When an exchange of silence process begins, nodes in both two sides of the link start a new handshake attempt cycle, keep silent for T_{Silence} first, and then begin to send NULLs. However, if L_{NULL} is shorter than the length of fiber optic link L, side A will go into silence and wait for another handshake cycle before it receives the NULLs sent by side B in this handshake cycle, which will cause handshake failure. In other case, when side A goes into silence and starts a new handshake cycle, side B will start a new handshake cycle and go into silence too after it receives the silence sent by side A. In this condition, if the link is not cleared(means that not both direction of the link is filled up with silence) within the time of T_{Silence}, the characters of previous transmission of handshake cycle which are left in the link will disturb the new cycle of handshake, causing link failure.

Therefore, the condition of stable handshake is: (1) L_{NULL} should be longer than L; (2) When a new cycle of handshake begins, both direction of the link should be cleared before the new NULLs is sent. In mathematical expression is:

$$\frac{L_{NULL} > L}{L_{Silence} > 2L} \Rightarrow L < min(\frac{c}{n}T_{NULL}, \frac{c}{2n}T_{Silence})$$

In the condition of $T_{Silence}$ =6.4us+12.8us and T_{NULL} =12.8us, assuming that the refractive index of the SMF is *n*=1.462^[13], the result is L<1970m, means that with the original parameter of SpaceWire protocol, the Optical SpaceWire can achieve stable handshake at the maximum distance of 1970m.

B. Flow Control Mechanism

The SpaceWire Flow Control mechanism prevents N-Char(valid data byte) from overflow. SpaceWire Codec should calculate both local credit count C_{credit} and remote credit count C_{outstanding}. When a SpaceWire Codec is ready to receive *F* more N-Chars, it sends one FCT, and C_{outstanding} adds *F*. When it receives one FCT, C_{credit} adds *F*, indicating that it is permitted to transmit *F* more N-Char. And the maximum value of C_{credit} and C_{outstanding} is β ·*F*. According to SpaceWire protocol, β =7 and *F*=8.

In long fiber, there may appear to be discontinuities between N-Chars due to the propagation delay, it is shown in Fig.5. For instance, both sides hold C_{credit} for $\beta \cdot F$ after link initialized, side A begins to send N-Char and it C_{credit} starts to fall. When side B receives $F \times N$ -Char and passes them to the application layer, it will send FCT to permit side A to send more N-Char. However, due to the propagation delay, C_{credit} of side A may falls to 0 before it receives FCT sent by side B. So side A will wait for the FCTs and send NULLs to maintain the link, causing N-Char discontinuity, decreasing the effective data rate.



Fig. 5. Propagation of FCT/N-Char in Long Fiber

Therefore, the condition of achieving full-rate transmission is that, C_{credit} should be ensure not falling to 0 before the Codec receive FCT from the other side. If the SerDes reference clock frequency is f Hz, it costs 1/f s for side A to send one character, C_{credit} falls from $\beta \cdot F$ to 0 will cost $\beta \cdot F/f$ s. The first N-Char sent by side A will arrive at side B in L/(c/n)=nL/c s. Side B will start to send FCT after receiving $F \times N$ -Chars, it costs F/f s. The first FCT sent by side B will arrive at side A in L/(c/n)=nL/c s. Thus the condition of fullrate transmission is:

$$\frac{nL}{c} + \frac{F}{f} + \frac{nL}{c} < \frac{\beta F}{f} \implies L < \frac{(\beta - 1)Fc}{2fn}$$

Assuming that the full-rate transmission distance is $L_0=[(\beta-1)Fc]/(2fn)$. When $L>L_0$, as it is illustrated in Fig.5, there will be NULLs with the length of $L_{NULL}=2(L-L_0)$ between every continuous N-Chars with the length of $L_{N-Char}=(\beta \cdot F/f) \cdot (c/n)$. Assuming the maximum data rate(there is no discontinuities between N-Chars) is R_{max} , the actual data rate

is R, and assigning $R=\lambda R_{max}$ (λ indicates the Bandwidth Utilization). Indeed, in continuous data transmission, assuming the amount of N-Char is N_{N-Char}, the amount of the total characters is N_{Total}, there is:

$$\begin{cases} \lambda = \frac{N_{N-Char}}{N_{Total}} = \frac{L_{N-Char}}{L_{Null} + L_{N-Char}} \\ L_{N-Char} = \frac{\beta F}{f} \cdot \frac{c}{n} \\ L_{Null} = 2(L - L_0) \\ L_0 = \frac{(\beta - 1)Fc}{2fn} \end{cases} \Rightarrow \lambda = \frac{\frac{\beta Fc}{fn}}{\frac{Fc}{fn} + 2L}$$

Then, the relationship between transmission distance and actual data rate (L– λ model) can be concluded as:

$$R = R_{max} , \text{ When } L \le L_0$$

$$R = \frac{\beta Fc}{\frac{fc}{fn}} R_{max} , \text{ When } L > L_0$$

In this model, *c* is vacuum light speed, *n* is the refractive index of the fiber, *f* is SerDes reference clock frequency, R_{max} is the full data rate related with *f* and SerDes internal PLL. These parameter are determined when the physical layer is established and can be considered as constant. $\beta \times F$ is the maximum number of FCT multiply by the number of N-Char per FCT, L is transmission distance, and they are variable.



Fig. 6. L- λ models with $\beta \times F=7 \times 8, 7 \times 16, 7 \times 32, 15 \times 32, 15 \times 64, 31 \times 64, 31 \times 128$

Fig. 6. is the L- λ model in the condition of f=100MHz, $c=3\times10^8$ m/s, n=1.462, but with different $\beta \times F$. It shows that with the larger value of $\beta \times F$, the Optical SpaceWire link can achieve longer full-rate distance L₀ and higher data rate in long distance transmission. With the parameter of original SpaceWire protocol, $\beta \times F=7\times8$, the Optical SpaceWire can achieve full data rate at the maximum distance of 49.245m, and the actual data rate will decrease sharply with the continuing increase of transmission distance.

V. EXPERIMENT

The experimental Optical SpaceWire point to point data transmission system is shown in Fig.1. and Fig.7. The

experiment includes full data rate test and $L-\lambda$ model verification.

Actually, the maximum handshake distance(1970m) of original SpaceWire $T_{Silence}$ and T_{NULL} is enough for most of the data acquisition system, but in the experiment, the fiber optic is longer than 7km, the original $T_{Silence}$ and T_{NULL} of SpaceWire handshake state-machine is unable to achieve stable handshake. Therefore, considering both propagation delay and SerDes CDR^[14] time, $T_{Silence}$ and T_{NULL} of the Optical SpaceWire Codec implemented in this paper is adjusted for $T_{Silence}$ =64us+128us and T_{NULL} =128us, which is able to achieve stable handshake in 19.7km long fiber optics.

To increase the full-rate distance, the value of $\beta \times F$ in Optical SpaceWire Codec is adjusted to15×64, which need 1024 bytes Tx/Rx buffer size. According to the reference [15], the resource usage will not increase obviously with the increase of buffer size when SpaceWire codec is implemented on FPGA, if buffer size is within 1024,

There are two fibers in the experiment: a 3m fiber is for full-rate test and a 7364m fiber is for L- λ model verification. The condition of the experiment is *f*=96MHz, providing 1.92Gbit/s link rate and theoretically 1.536Gbit/s full data rate (80% 8b/10b code efficiency). The RI of the 7364m fiber *n* is calculated through Time-Code loop test, the result is *n*=1.463. The full-rate test with 3m fiber is shown in Fig.7.



Fig. 7. Full-rate Test.

The SerDes data width is 16bits, so the software statistic actually shows the number of 16bits wide Words. The test result data rate is R'_{max} =93.88MWord/s, so the actual full data rate is R_{max} =2R'_max=187.76MByte/s=1.5Gbit/s.

Then the 7364m long fiber is used to verify the $L-\lambda$ model. The result is shown in Fig.8.

With the 7364m fiber, the test result shows the tested data rate is $\dot{R}=13.17MW$ ord/s, thus the actual data rate is R=2R=26.34MB yte/s, and $\lambda=R/R_{max}=0.140$.

According to the L– λ model, in the experiment condition, when L=7364m, it can be calculated that λ_c =0.138, R_c= $\lambda_c R_{max}$ =25.91MByte/s. The result shows that the data rate in long fiber calculated by the L– λ model meets the actual tested data rate, with an error of σ =(R–R_c)/R=1.6%.

Therefore, it can be considered that according to the L- λ model, the original SpaceWire link with $\beta \times F=7 \times 8$ can achieve full-rate transmission at 51m distance, while the Optical

SpaceWire link in this paper can achieve full-rate transmission at 957m distance. And with 7364m fiber, the Optical SpaceWire with original $\beta \times F=7 \times 8$ can achieve 1.52MByte/s data rate, while the Optical SpaceWire with modified $\beta \times F=15 \times 64$ can achieve 26.34MByte/s data rate.



Fig. 8. Long Fiber Data Rate Test

VI. CONCLUSION

In this paper, an Optical SpaceWire point to point transmission system is implemented, which uses fiber optics instead of electrical cable as transmission media. The Optical SpaceWire link provides 1.92Gbit/s link rate, and achieves 1.5Gbit/s full data rate which is approximately 10 times of the original SpaceWire link. With the parameters in the handshake and flow control mechanism modified according to the model built in this paper, it can also achieve full-rate at more than 900m transmission distance, which is about 19times of the SpaceWire with the original parameters.

The SpaceWire fiber optic data transmission system implemented in this paper gives a solution of higher-speed and long-distance SpaceWire data link, which can expand the application prospect of the SpaceWire.

REFERENCES

- ECSS-E-ST-50-12C SpaceWire-Links, nodes, routers and networks[S]. Noordwijk, The Netherlands: European Cooperation for Space Standardization, 2008: 13-14.
- [2] Steve P, Albert F. SpaceWire-RT[C]// International SpaceWire Conference Nara 2008 Conference Proceedings. Dundee, UK: Space Technology Centre, University of Dundee, 2008: 15-23.
- [3] Evgeny Y, Yuriy S, Elena S, et al. GigaSpaceWire Gigabit Links for SpaceWire Networks[C] // Proceedings of the 5th

International SpaceWire Conference. Dundee, UK: Space Technology Centre, University of Dundee, 2013: 28-34.

- [4] Michael E, Steven T. Galvanically Isolated SpaceWire[C]// Proceedings of the 6th International SpaceWire Conference. Dundee, UK: Space Technology Centre, University of Dundee, 2014: 113-118.
- [5] Steve P, Chris M, Albert F, et al. SpaceFibre: Multiple Gbit/s Network Technology with QoS, FDIR and SpaceWire Packet Transfer Capabilities[C]// Proceedings of the 5th International SpaceWire Conference. Dundee, UK: Space Technology Centre, University of Dundee, 2013: 11-18.
- [6] Tetsuya K, Atsushi K, Yuki Y, et al. Impact of wave propagation delay on latency in optical communication systems[C]// Proc. SPIE 8646, Optical Metro Networks and Short-Haul Systems V, 86460C (December 26, 2012). San Francisco, California, USA | February 02, 2013. DOI: 10.1117/12.1000190.
- Steve P, Chris M, David M, et al. SpaceWire and SpaceFibre on the Microsemi RTG4 FPGA[C]// Proceedings of the 2016 IEEE Aerospace Conference. 2016: 1-8. DOI: 10.1109/AERO.2016.7500644.
- [8] Bruce Y, Alberto G, Albert F, et al. Integrating STAR-Dundee SpaceFibre Codec with TI TLK2711[C]// Proceedings of the 6th International SpaceWire Conference. Dundee, UK: Space Technology Centre, University of Dundee, 2014: 249-252.
- [9] Vladimir G, Dmitri S, Mikhail M, et al. Advanced Oversampling Techniques for the SpaceFibre[C]// Proceedings of the 6th International SpaceWire Conference. Dundee, UK: Space Technology Centre, University of Dundee, 2014: 240-243.
- [10] Sachin H, Sudhir D. Open Loop and Closed Loop solution for clock domain crossing Faults[C]// Proceedings of 2015 Global Conference on Communication Technologies. 2015: 645-649. DOI: 10.1109/GCCT.2015.7342741.
- [11] Mohammand M. An 8b/10b Encoding Serializer/Deserializer (SerDes) Circuit for High Speed Communication Applications Using a DC Balanced, Partitioned-Block, 8b/10b Transmission Code[J]. International Journal of Electronics and Electrical Engineering, 2015: 3(2): 144-148.
- [12] Yu Z, Hu Q. A Comma Detection and Word Alignment Circuit for High-speed SerDes[C]// Proceedings of the 2011 7th International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM). 2011: 1-4. DOI: 10.1109/wicom.2011.6040215.
- [13] Vjaceslavs B, Sandis S, Girts I. Latency causes and reduction in optical metro networks[C]// Proc. SPIE 9008, Optical Metro Networks and Short-Haul Systems VI, 90080C (December 20, 2013). San Francisco, California, United States | February 01, 2014. DOI:10.1117/12.2041736.
- [14] Kareem I, Tawfik I, Hassan M. Design and Implementation of CDR and SerDes for High Speed Optical Communication Networks Using FPGA[C]// Proceedings of the 2016 18th International Conference on Transparent Optical Networks (ICTON). 2016: 1-3. DOI: 10.1109/ICTON.2016.7550487.
- [15] Matthew R, Martin S. An Experimental Evaluation of SpaceFibre Resource Requirements[C]// Proceedings of the 6th International SpaceWire Conference. Dundee, UK: Space Technology Centre, University of Dundee, 2014: 228-232.

Development of the new Mission Data Recorder (MDR) with SpaceWire interface onboard the ERG spacecraft and the result of its performance on orbits

On-board equipment and software, Short Paper

Sadatoshi Eguchi Institute of Space and Astronautical Science (ISAS) Japan Aerospace Exploration Agency (JAXA) Sagamihara-City, Kanagawa-Pref., Japan eguchi@stp.isas.jaxa.jp

Abstract-The Mission Data Recorder (MDR) has been developed and currently operated for ERG mission that is exploration of radiation belts around the earth. The MDR consists of 32GB Flash-Memories and 128 MB SDRAMs. The MDR record system is similar to a kind of file system. This file system supports (1) multi-partitions; the system manages maximum to 64 partitions, (2) ware levelling; that function extends Flash-memory's life time, (3) bad-block management for Flash-memories and (4) time search access; the system manages stored data with "time-tag" in order to time search and random access for stored data by "time" instead of memory pointers. Users can obtain necessary data from the MDR only to use API functions for time-tag search. During ERG operation, the MDR processes over 5GB data which are received via SpaceWire mission-network in a day. It works well under strong radiation environment in Van Allen belts although single event upsets are detected and corrected once a few days.

Index Terms—SpaceWire, On-board equipment, Flash-ROM, filesystem, High-Reliability

I. INTRODUCTION

In the ERG satellite, high time resolution data is collected, and the collected data amount is 5 GB on average per day. Operation for selecting event data from this 5 GB data and downloading it to the ground station is carried out. Mission Data Recorder (MDR) is the device that temporarily stores 5 GB of data. MDR consists of 32 GB of flash memories, 128 MB of SDRAM and installed software. It is also designed to save input data of 11 Mbps to the flash memories from the mission request. The data stored in the flash memories can be read by onboard applications on MDR. Applications process read data and outputs them as telemetry packets.

In this paper, MDR data processing system and error handling are introduced. In addition, we will also introduce the results of operation in past operations. Takeshi Takashima, Iku Shinohara Institute of Space and Astronautical Science (ISAS) Japan Aerospace Exploration Agency (JAXA) Sagamihara-City, Kanagawa-Pref., Japan ttakeshi@stp.isas.jaxa.jp, iku@stp.isas.jaxa.jp

II. RECORD SYSTEM

The MDR record system is similar to a kind of file system. This file system supports 5 functions as follows.

A. Multi-partitions

the system manages maximum to 64 partitions. It is possible to logically divide the Flash-Memories by setting the following information in the table, which is called "Partition Management Table".

- APID and Category (ID for identifying partition)
- Start address and End address
- First pointers for "Stored Data Table" (refer III.)

Space Packet is adopted as the upper layer protocol of SpaceWire / RMAP in MDR received data format [1]. When the MDR receives the data, it reads the APID and the Category included in the Space Packet Header and checks whether there is a corresponding partition. When there is a partition, receive data information is registered in a queue prepared for each partition. One page of data is accumulated in the queue, or when a certain time has elapsed, DMA transfer is performed and data is written from the receiving buffer to the Flash-Memories.

B. Bad-block Management

In FLASH-Memories, bad blocks (blocks in which data can not be written) are generated due to element deterioration or the like. There are two types of bad blocks, one that is already bad block at the time of shipment and one that becomes bad block due to write life. Both of these bad blocks are managed with one bad block table.

Blocks that are already bad blocks when shipped are registered in the bad block table beforehand.

Blocks that have reached the end of write life fail to write to Flash-ROM. If data writing fails, it is registered in the bad block table, and data is written to the next block.

C. Ware Levelling

This function extends Flash-memory's life time. Input data is written in order from the start address to the end address of each partition area, so that the number of times of writing is made uniform and the life of Flash-memories are extended.

When the block to be written is a bad block or a protected block, it is not written to the corresponding block and written to the next block.



Fig. 1. MDR Ware Levelling

D. Time Search Access

The MDR manages stored data with "time-tag" in order to time search and random access for stored data by "time" instead of memory pointers. User applications can obtain data from the MDR by using API functions for time-tag search.

In addition, since the storage information table has a list structure for each partition, time range search is also possible.



Fig. 2. MDR Data Management Overview

E. Application Program Interface(API)

To support data access by onboard user applications, API functions are provided in MDR. TABLE I. shows key API functions.

TABLE I. API FUNCTIONS FOR MDR

API functions	Description
Write Protect	Overwriting the stored data is prohibited,
	and data can be read.
	Multiple data corresponding to the specified
	time range can be processed collectively.

API functions	Description
Data Search	Search stored data. When the search target
	time is specified, the closest time tag is
	returned.
Data Shift	Returns the time tag before (or after) the
	specified number based on the data
	corresponding to the time tag acquired by the
	Data Search API.
Data Get	Data corresponding to the time tag acquired by the Data Search API or Data Shift API is read from Flash-Memories and copied to the specified address.
Data Release	Cancels overwriting prohibition of saved data and erases data. Data corresponding to the specified time range can be erased collectively.

III. RECORD PROCESSING

The processing that MDR stores data received from SpaceWire network in flash memory is as follows.

A. Memory

The MDR is equipped with EEPROM, SRAM, SDRAM and flash memory, and it is used in the following way.

TABLE II.	API FUNCTIONS FOR MDR
-----------	-----------------------

Memory	Description
EEPROM	Program
SRAM	Error information recording
SDRAM	Work area of software
	Send/Receive buffer for SpaceWire network
	Flash-Memory management area
	Bad block information
Flash-	Data recording
Memory	Bad block information (backup)
	Part of Flash-Memory management area
	(backup)

B. Flash-Memories Management Data

In the MDR, a management table for recording information such as the write address of the data written in the flash memory is prepared. This is called "Stored Data Table". MDR receives write data divided in 2 kilo byte units and writes it to the Flash-Memories while being divided. MDR records the information required for reading data, such as the write address of the flash memory and time-tag, in this Stored Data Table.

The operation of Stored Data Table is shown in Fig. 3. and the following.

Empty list: In the initial state, the Stored Data Table is all empty state, and it is one list.

Writing list: When MDR receives the first segment packet, it retrieves one element from the Empty list. The element is updated with all information from the first segment packet to the last segment packet.

Non-Protected List: When all segmented packets are stored in Flash-memories, the corresponding elements are moved to the "Non-Protected" list. Data registered in the "Non-Protected" list is deleted in order of oldest when data is recorded in all areas of the partition.

Protected list: When the "Write Protect" API function is called by the application, target elements are moved to "Protected" list. MDR permits access only by data registered in this protected list. Elements on the "Protected" list will not be deleted with data on Flash-Memories even if data is recorded in all areas of the partition.

Release: Moving an element to the Empty list is called "Release". The target elements to be freed are: "Write Protect" When an API function is called, it is older than the target element. When the "Release" API function is called by the application, the target element.

The "Writing" list, the "Non-Protected" list, and the "Protected" list are prepared for each partition and can be referred to from the "Partition Management Table".



Fig. 3. MDR Stored Data Table

C. Record Process

MDR receives recording data from the SpaceWire network in the form of RMAP Write command. When the received RAMP Write command is normal, the data part of the RAMP Write command is DMA transferred to the reception buffer, which is allocated in the SDRAM.

The format of the RMAP Write command data is the Space Packet format. MDR determines the recording destination partition from the header information of the Space Packet and records the received data in the Flash-Memory in the divided state while keeping the Space Packet format. The address of the recording destination is registered in the Stored Data Table together with the "time-tag" of the Space Packet header and used for Time Search Access.

Combining received data is done using the Space packet header when the on-board application calls the Data Get API function and reads the recorded data from the flash memory.

IV. ERROR HANDLING

In order to detect anomalies in received data caused by factors such as radiation or SpaceWire network failure, MDR

has a function to detect errors in received data. MDR uses the function of RMAP / Space packet which adopted the data received from the SpaceWire network as the upper protocol of SpaceWire and guarantees the soundness of the received data. The presence or absence of data abnormality is recorded in the management table and can be discriminated from the on-board application.

Error handling functions by MDR are shown below.

A. 1bit/2bit Error Detection

MDR mounted flash memory, SDRAM and FPGA memory has 1 bit error correction / 2 bit error detection function. The transmission FIFO and the reception FIFO which are SpaceWire I/F also have a 1-bit error correction / 2-bit error detection function.

Memory patrol periodically operates for SDRAM and SRAM. If 1 bit error is detected, data corrected by ECC is written to the corresponding address and returned to normal data. It is almost impossible for a 2-bit error to occur during operation due to memory patrol.

B. Staying Error Packet Detection

When a Staying Error Packet occurs in the mission network, error detection, separation, and recovery are performed automatically by the Real-Time Control function. MDR marks the management table corresponding to the packet as abnormal data.

C. RMAP Packet Error Detection

Checking by CRC and checking of RMAP header are performed according to the RMAP protocol. When an abnormality (including a CRC error) is detected in the RMAP header, the abnormality content is recorded and the RMAP packet is discarded, and the process immediately returns so that the next packet can be received. When an abnormality is detected in the RMAP data CRC or when the EEP termination is made, the contents of the abnormality are recorded and immediately returning so that the next packet can be received. The packet is not discarded, but the corresponding management table is marked as abnormal data.

D. Space Packet Error Detection

The sequence counter is checked, if there is duplication, error information is recorded, the packet is discarded, and the processing proceeds to the next data processing. If the sequence counters are not continuous, error information is recorded but the packet is saved in the flash memory and marked as abnormal data in the management table. Also, it checks whether the APID of the primary header and the Category ID (partition ID) of the secondary header correspond to the MDR partition. If it does not correspond, record error information and discard the packet, proceed to the next data processing.

V. PERFOMANCE

TABLE III. is the amount of data processed by MDR. TABLE IV. is statistics of SEU errors detected by MDR. MDR works well under strong radiation environment in Van Allen belts although single event upsets are detected and corrected once a few days.

TABLE V. is statistics of error packet count detected by MDR. MDR has processed all errors appropriately, so that the application software on MDR can recognize data abnormality.

TABLE III. PERFORMANCE OF MDR

Item	Amount from 2017/1/10 to 2017/11/30		
Operation time	319 days		
Data stored into Flash-Memory	250.4GB		
Data loaded from Flash-Memory	140.4GB		
Stored Data rate (average)	112kbps		
Stored data rate (maximum)	7.53Mbps		

TABLE IV. SEU ERROR COUNT

Memory	Expect	Result
Flash-	1bit error: Once in 4.5m	1bit error: Once in 81.7m
Memory	2bit error: Once in 1.8y	2bit error: (none)
CDAM	1bit error: Once in 29.9h	1bit error: Once in 7.0H
SRAM	2bit error: Once in 4.1*10 ⁵ y	2bit error: (none)
CDDAM	1bit error: Once in 175d	1bit error: (none)
SDRAM	2bit error: Once in 7.6*10 ⁹ y	2bit error: (none)
FPGA	1bit error: Once in 1.7d	1bit error: (none)
Internal	2bit error: Once in 2.6*10 ⁶ y	2bit error: (none)
RAM		. ,

TABLE V. ERROR PACKET COUNT

Error	Result
Staying Error Packet	(none)
RMAP Packet Error	1
Space Packet Error	(none)
Space Packet Discontinue	16

REFERENCES

- CCSDS Secretariat Office of Space Communication (Code M-3) National Aeronautics and Space Administration, "SPACE PACKET PROTOCOL" CCSDS 133.0-B-1 Cor. 2, September 2012.
- [2] ESA Requirements and Standards Division, "ECSS-E-ST-50-52C Space engineering - SpaceWire - Remote memory access protocol," 5 February 2010.

Communication Protocol Based on Time Multiplexing for SpaceWire Data Handling Networks

SpaceWire Networks and Protocols, Short Paper

Yu Junhui, Niu Yuehua, Li Xiaojuan Beijing Institute of Spacecraft System Engineering China Academy of Space Technology (CAST) Beijing, China Yujunhui225@163.com

Abstract—A communication protocol based on time multiplexing (TMCP for short) is proposed in this article. TMCP is designed to provide services for high speed payload data as well as time-critical data on a single network. In order to guarantee deterministic data delivery for time sensitive data and efficient transfer for high speed payload data, three different services with different priorities are proposed. A host is needed in the protocol, which is responsible for the dynamic bandwidth allocation over the network. Other nodes in the network make all the transfers following the indication of the host, without necessarily maintaining any schedules, which may show its merit in the future Plug and Play networks. Remote Memory Access Protocol (RMAP) is used in TMCP to make basic communication.

Index Terms—SpaceWire, Network, Time Multiplexing, Communication Protocol.

I. INTRODUCTION

SpaceWire has emerged as a high-speed (2-400Mbps), low-power and low-cost network for spacecraft [1]. Its wormhole routing routers make the transmission delay on SpaceWire networks unpredictable [2].

Time multiplexing and scheduling are used in network technologies to obtain guaranteed transmission characteristics. Approaches have been made based on time multiplexing. SpaceWire-D, for example, is a standard prototype providing deterministic data delivery over SpaceWire using a scheduling mechanism [3]. It divides the operation time into different slots and a schedule table is built defining in which slot an initiator is allowed to send data. According to this protocol, each node in the network should hold a copy of this schedule table and build error detection and recovery mechanism accordingly. The schedule table during a mission is statically configured so that it doesn't apply to dynamically configured environment.

There are two types of data that need to be handled on a spacecraft: the platform control data, which are time sensitive and normally transferred between the central computer of the data handling system and other terminals onboard, and the payload data which need greater bandwidth but have better tolerance of time delay. SpaceWire should meet the transfer needs of both types of data to become the sole data handling network onboard [4].

Based on the analysis above, a communication protocol based on time multiplexing (TMCP) is introduced in this article to provide services for both avionics data and payload data handling on a single SpaceWire network. In our approach, there should be only one unique host (usually the central computer of the onboard data handling system) in the SpaceWire network and it is responsible for the bandwidth allocation over the entire network. TMCP provides three services with different priorities for different kinds of data traffic onboard. Static data service supports pre-scheduled data transfers of restricted length between the slave nodes and the host without handshake. This service has the highest priority and can be used for time sensitive and periodic data transfer. A typical application scenario is for regular telemetry collection and important telecommand distribution. Burst data service has the secondary priority. This service supports data transfers between the host and other nodes on request of the initiator. And bulk data service supports high speed data transfers over the network, which is usually applied in payload data transfer.

TMCP divides the operation time into several frames. During each frame, data transfers applying static data service, burst data service and bulk data service are performed in priority order. TMCP uses the SpaceWire Remote Memory Access Protocol (RMAP) [5] to make basic communication. All the data transfers under this protocol are triggered by the network host. The host is capable of building a more flexible and reasonable bandwidth allocation over the network based on the requests and the responses of the data services from other nodes. The slave nodes do not necessarily need to keep the schedule table, which gains advantages in unified design and dynamic network configuration.

II. PROTOCOL STACK

The protocol stack of TMCP is illustrated in Fig.1. There are two types of devices on SpaceWire networks applying TMCP: the host and the slave nodes. It is the host who is responsible for initiating all the data transfers and it is unique over the network. The host is usually the central computer of

the onboard data handling system and the salve nodes can be other terminal controllers or massive memory, instrument, sensor etc.



Fig. 1. TMCP Protocol Stack.

TMCP runs over RMAP, and it provides three services with different priorities: static data service, burst data service and bulk data service. The first two kinds of services only support data transfers between the host and other nodes, while the third service supports data transfers between the slave nodes.

- Static Data Service: has the highest priority. Data transfers using this service are prescheduled in one or more specified frames. There may be multiple data transfers using this service in one frame, and the transfer order is set by the host. It provides the most deterministic service.
- Burst Data Service: has the secondary priority. User • application making data transfers with this service should first bring up a transfer request with data amount to the host, and the host would initiate the transfer in the next frame with abundant free time.
- Bulk Data Service: has the lowest priority. It provides services for data transfers between the slave nodes over the network. During each frame, one slave node would be notified by the host to make data transfers within certain time scope.

III. THE DESIGN OF TMCP

The working mechanism and different services offered by TMCP are explained in this chapter.

A. Communication Frames



TMCP is a communication protocol based on time multiplexing, which provides a deterministic way of data transferring. The communication over the network is divided into a sequence of communication frames by the broadcasting of time code. Communication frame N is started by time code N and is ended by time code N+1, as illustrated in Fig.2.

The duration of a communication frame should be set based on the implementation of the system. Typically, during each communication frame, data transfers applying the three services are made in order, as shown in Fig.2. The time window needed by static data service is pre-allocated while the time windows of the other two services are distributed by the host dynamically. To make sure that all the transfers in a frame have completed when the next time code comes, it is suggested that a free window should be left at the end of the frame.

B. Data Transfers

There are three types of data transfers over the network applying TMCP: data transfer from the host to other nodes, data transfers from other nodes to the host and data transfers between the other nodes. The data transfers between the host and other nodes are conveyed by writing or reading RAMP command initiated by the host. The data transfers between the slave nodes can only happen within the Bulk Data Service time window, and they are sent by the node which is provided with the time window using RAMP commands as well.

C. Static Data Service

Static data service guarantees the most deterministic data transfers. The time window that static data transfers need is pre-allocated in the communication frames, which is right after the broadcasting of the time code and at the beginning of the communication frames. To make this work, the host should hold a schedule, which contains information about reading or writing (getting or sending) how much data from/to which node in which frames. This schedule should be set based on the data transfer need of the system and it is static during the mission. During one communication frame, multiple static data transfers could be made. And there may be no static data transfers in one frame as well. The slave nodes on the other hand, do not necessarily need to keep such schedule since it is the host who initiates the data transfers.



Fig.3. Data Transfers with Static Data Service.

The process of data transfers with static data service is illustrated in Fig.3. The data transfer should be submitted by the host application to the service provider before the due communication frame. Then the transfer would be made in form of a reading or writing RAMP command when the prescheduled time window comes. The host application should be mentioned at the end of the transfer.

D. Burst Data Service

The time window of burst data service comes after the static data service in a communication frame. It may begin right after the time code if there is no static data transfer prescheduled in the current frame. The time windows of this service vary in each frame based on the service requests that the host service provider collects. During each frame, the host should collect the service requests made by its application and poll the requests from the slave nodes. The transfer information should be in the request, including the transfer type, the target node and the time window needed for the transfer. The host would schedule all these transfers at the beginning of the next frame as long as enough free time on the net. If there is not enough time for all the transfers, the host should decide the transfer order based on the implementation of the system and leave the unscheduled transfers to the next frame.

The process of data transfers required by the host application using burst data service is almost the same with static data service. The host application should first bring up with the service request containing information of the transfer and the service provider of the host would make schedule for the transfer in the next accessible communication frame.



Fig.4. Data Transfers with Burst Data Service.

The process of data transfers required by the slave node application using burst data service is illustrated in Fig.4. The slave node application should first submit its service request to the service provider and wait for the request to be polled by the host. The host would schedule the transfer in the next accessible frame and when the time window for the transfer comes, the data transfer would be made in form of RMAP reading command. The slave node application should be informed once the transfer has been completed and the data transferred should be delivered to the host application for further use.

E. Bulk Data Service

The host should calculate the time window left for the bulk data service after scheduling the static and burst data transfers at the beginning of each frame and decide to which node should this free time window be distributed to. The host may hold a distribution table to help itself make the dicision. The slave node selected could initiate data transfers with other nodes within the time window it is informed with. The initiator of these tranfers is usullay payload control processor or other teminal controller, while the target is usually memory, sensor or instrument on the network. It is suggested that the host leaves a small free time window at the end of the frame to make sure that all the transfers have been completed when the next time code comes.



Fig.5. Data Transfers with Bulk Data Service.

The data transfer process with bulk data service is quite simple, as shown in Fig.5. The slave node picked would be informed with the length of the time window, and it could initiate all the data transfers within the window. Multiple times of data transfers may be initiated. But the initiator should confirm that all the transfers must be completed before the time window closes. A confirmation should be sent to the host after the transfers are completed.

F. Fault Detection

In TMCP, it is important that all the data transfers should be done within the communication frame that they are scheduled in so that the transfer delay of time-critical data is deterministic. Thus, it is suggested that a small free window should be left after the bulk data service. The on-going transfer should be halted immediately when a time code is broadcasted to the node and an error notification should be brought up to the node's application. An error notification should be sent to the host application as well, if the bulk data service confirmation has not been received before the next time code comes. The user application should decide the appropriate reaction to the errors.

IV. MAIN INNOVATIONS

The main innovations of TMCP are listed as following:

- An exclusive host is defined in the protocol, which is usually the central computer of the onboard data handling system. Unlike SpaceWire-D [3] in one initiator mode, the slave nodes in TMCP could be payload control processors as well as instruments, sensors or actuators.
- Three services with different priorities are provided by TMCP. Static data service and burst data service are mainly used for avionics data while bulk data service is used for high speed data. It is designed for SpaceWire to be the sole network on board.
- The time window of static data service is preserved while the time windows of the other two services are dynamically distributed by the host, which may lead to a better allocation of the bandwidth on the network.
- All the transfers in TMCP are triggered by the host (even in bulk data service, the initiator can only begin the transfers once it is informed) so that the slave nodes do not need to keep up with the schedule. This may show advantages in unified design and dynamic network configuration in the future.

V. CONCLUSION

TMCP is designed based on the onboard data analysis. Time multiplexing mechanism and services with different priorities are introduced to provide deterministic transfer delay for time-critical data and abundant bandwidth for high speed data. An exclusive host is defined in the protocol and it is responsible for the bandwidth allocation over the whole network. TMCP supports common avionics data and payload data handing on a single SpaceWire network and it contributes to a unified design in the network nodes and the future Plug and Play networks.

REFERENCES

- [1] S. Parkes, P. Armbruster. "SpaceWire: Spacecraft onboard data handling network," Acta Astronautica, 66, 88-95, 2010.
- [2] ECSS-E-ST-50-12A, Space Engineering SpaceWire Links, nodes, routers and networks. 24 January 2003.
- [3] S. Parkers, D. Gibson, A. Ferrer. "SpaceWire-D: Deterministic Data Delivery over SpaceWire", Data Systems in Aerospace, Warsaw, Poland, June 2014.
- [4] Wang Zhen, Lu Guoping. "Spaceborne unified data &information network", 2014 International SpaceWire Conference, September 2014.
- [5] ECSS-E-ST-50-52C, Space Engineering SpaceWire Remote memory access protocol. 5 February 2010

SpaceVNX: A Scalable and Highly Flexible Small Form Factor Standard for Small-Sats and Cubesats

PAPER NOT PUBLISHED

Radiation tolerant microprocessor for the Computer Vision with SpaceFibre links

Components, Short Paper

Tatiana Solokhina, Jaroslav Petrichkovich, Alexander Glushkov, Leonid Menshenin, Denis Kuznetsov, ELVEES RnD Center, JSC, Zelenograd, Russia, tanya@elvees.com Sheynin Yuriy, Suvorova Elena University of Aerospace Instrumentation, St. Petersburg, Russia sheynin@aanet.ru Steve Parkes, Space Technology Centre, University of Dundee, 166 Nethergate, Dundee, DD1 4EE, UK sparkes@computing.dundee.ac.uk

Dmitri Dymov, Information Satellite Systems (ISS) – Reshetnev Company, Zheleznogorsk, Russia dymovdv@mail.ru

Abstract — The article presents a Radiation tolerant ASIC as the SoC (System-on-Chip) with Computer Vision (CV) HW and built-in multichannel multiprotocol SpaceFibre, GigaSpaceWire (SpaceWire-RUS standard), SpaceWire based switch for the space applications.

Index Terms — Radiation tolerant heterogeneous Multicore ASIC, Computer Vision, CNN, SpaceFibre SpaceWire, GigaSpaceWire, RMAP and STP-ISS Transport protocols

I. INTRODUCTION

In the space "smart" cameras with video processing and storing systems it is necessary to solve several important tasks, including:

- 1. Delivering large amounts of video data from the high speed video sensors to the proper processing system;
- 2. High-speed data streams switching;
- 3. Storing large amount of video data;
- 4. Recognition and deep learning;
- 5. The overall management of space system.

The article describes the aspects of CV ASIC development as the "system-on-chip» qualified for space applications with architecture intended to provide high-speed data exchange between video data source, video data processing and video data storage blocks, and also between multiprocessor networks on the SpaceFibre, SpaceWire with STP-ISS transport protocol, and Giga SpaceWire (SpaceWire-RUS standard) base.

II. CV ASIC ARHITECTURE

The 90nm CMOS MCDE50F microprocessor ASIC is heterogeneous Multicore ASIC, based on a ELVEES Radiation tolerant library. It includes dual core 64-bit CPU cluster and

programmable ELVEES Video Engine with HW accelerator for the Convolutional Neural Networks (CNN).

These resources will provide opportunities for building space "smart" cameras with recognition capabilities and deep learning to use in satellites and spacecraft systems, as well as in space robotics and in the planetary surface exploring.

The SoC design and architecture support Single-Event-Upset (SEU) fault-tolerance. The ASIC embedded networking subsystem provides multiple ports for high-rate interconnection with combination of SpaceWire/GigaSpaceWire (SpaceWire-RUS)/ SpaceFibre links. Input and processed data streams are transmitted via 3.125 Gbps two SpaceFibre links with built-in DMA controllers. Two SpaceWire links (ECSS-E-50-12C) provide data transfer bandwidth of 2-300 Mbps. The ASIC embedded networking subsystem based on the SpaceFibre/ SpaceWire/GigaSpaceWire provides a balance between external and internal data throughput.

Two 64-bit CPU cores perform the system management. Each CPU has 64-bit FPU coprocessor. CPU also has a memory management unit (MMU) based on the fully associative address translation buffer (TLB), the instruction cache (I CACHE) and the data cache (D CACHE).

The programmable ELVEES Video Engine (Elcore50 or E50) with HW accelerator for the Convolutional Neural Networks (CNN) is used for CV tasks and will provide opportunities for building Space "smart" cameras with recognition capabilities and deep learning in satellites and spacecraft systems, as well as in Space robotics and in the planetary surface exploring.

The ASIC has two DDR2 memory port (1600MB/s each). It supports DMA transfers between external I/O ports and external memory and has Multifunctional Buffed Serial Port (MFBSP) that can act as SPI, I2S, LPORT, GPIO interfaces. SpaceFibre 3,125 Gbps Serial I/O ports will be used for the onboard Network with the opportunity to work via fiber-optic transceivers. SpaceFibre can be effectively used for connection

with the external FPGA, which supports interfaces with video sensors, and contains ELVEES IP-cores as ISP (Image Sensor preprocessor) and an H264 codec (Fig 1).

All ASIC memory blocks including the register files in CPU are protected by Hamming code with single error correction and two errors detection.

Internal networking subsystem provides multiple ports for high-rate interconnection with combination of the SpaceFibre and SpaceWire/GigaSpaceWire links on the ASIC.



Fig.1 Example of the SpaceFibre interface used for the external RT FPGA with built-in ISP and H264 codec for the connection to video sensors and the MCDE50F processor.

To complete the video processing system, it is required to connect external ISP and hardware encoder to the MCDE50F (if microprocessor ASIC is realized in 90nm CMOS process). This can be implemented as ASIC or as FPGA. ELVEES has developed video preprocessing and video encoding IP blocks adapted for the FPGA design.

In case of realization of the device based on 65 nm CMOS Radiation Tolerant libraries the blocks of ISP and codec will be integrated in one chip with other blocks of MCDE50F microprocessor with 6.25 Gbps SpaceFibre links.

The main features of proposed ELVEES ISP IP-core are:

- support parallel and serial interfaces of CMOS sensors;
- up to 12 bit width;
- support LVTTL Bayer / Mono CMOS, BT 656 10 bit, MIPI CSI-2 and RAW 24 bit protocol;
- data formats can be RGB, YCbCr 4: 4: 4 (4: 2: 2), Bayer, Mono;
- maximum resolution is FullHD for video and up to 16

megapixels for images;

- pixel corrections;
- dynamic range corrections;
- color space conversions;
- gamma corrections;
- noise reduction.

For the video compression, good option is to use the base profile of H.264 standard. In the case of the minimum delays, it is recommended to use only I frames. Proposed video encoder supports:

- up to FullHD resolution of input;
- encoding up to 30 fps;
- adaptive per frame based rate control;
- up to 16 Mbps output stream;

III. MCDE50F MICROPROCESSOR BLOCK – DIAGRAM



Fig.2 MCDE50F microprocessor block-diagram

MCDE50F microprocessor ASIC consists of the following main units:

- CPU 64-bit SIMD/FPU dual CPU core with instruction and data caches of the first and the second level;
- ELVEES video Engine (E50) IP-core;
- GPU graphics processor unit;
- MPORT 64-bit external memory port;
- DMA direct memory access controllers for the respective ports and controllers;
- two 32-bit DDR2 ports;
- UART-asynchronous serial port;
- Two port switch SpaceFibre/GigaSpaceWire (SpaceWire-RUS), not less than 3,125 Gbps (for each port);

- Two SpaceFibre licensed controllers supporting four virtual channels each (not less than 3,125 Gbps);
- SWIC four duplex channels in accord with the SpaceWire standard (ECSS-E-50-12C) with a bandwidth of 2 to 300 Mbps each. Support for the RMAP (Remote Memory Access Protocol) Protocol and Transport Protocol STP-ISS-13;
- MFBSP 4 multi-purpose buffered serial ports (SPI, I2C, PORT, GPIO);
- Ethernet MAC 10/100 MHz controller;
- JESD204B controller, 1 lane;
- VPORT video output port;
- IRQ interrupt controller;
- PLL programmable frequency multiplier based on PLL;
- IT0, IT1-universal interval/real-time timers;
- WDT-watchdog timer.

Some additional important features for the microprocessor:

- Embedded software debugging tools (on CD) with JTAG port according to IEEE 1149.1;
- correction of single errors and detection of double errors for the internal and external memory based on the modified Hamming code;
- Power management;
- built-in BSR register (Boundary Scan Register);
- built-in DFT (Design for Test) tools);
- Linux operating system support;
- Ceramic package (CPGA-720 or CLGA-720).
 - IV. IMPLEMENTATION OF THE MCDE50F SPACEFIBRE, GIGASPACEWIRE, SPACEWIRE INTERFACES

Implementation of the SpaceFibre, GigaSpaceWire, SpaceWire interfaces for MCDE50F microprocessor is based on the use of three subsystems:

a) The SpaceWire network controller subsystem (SpWT);

b) SpaceFibre Routing Switch - SpFR (SpaceFibre Routing Switch);

c) Two licensed SpaceFibre IP subsystem.

a) The SpaceWire network controller subsystem SpWT has hardware implementation of the Transport protocols RMAP and STP-ISS, along with raw SpaceWire packets transfer. Hardware implementation of Transport protocols provides high-performance information transfer and offloads the central processor from the protocols' implementation overheads. Protocol STP-ISS performs the functions of the transport layer, operating on top of the SpaceWire standard (ECSS-E-ST-50-12C) Protocol stack and the SpaceWire-RUS standard draft. Protocol STP-ISS determines the transport and interaction of nodes in the network, formats of transmitted data and rules for the transfer of messages between on-Board SpaceWire network nodes.

b) SpFR supports two multiprotocol GigaSpaceWire (SpaceWire-RUS standard)/SpaceFibre ports.

The structure of SpFR includes the following blocks: AXI master (AXI4 Master Interface); AHB slave (AHB Slave Interface); Registers (control and status registers set); HardRMAP (Controller RMAP); Controller of Control Codes; Packet Controller and DMA; Router (SpFi Network Layer); Giga Bds1, Giga Bds2 (GigaSpaceWire Interface); pFi1, SpFi2 (SpaceFibre Interface); PMA (Physical Media Attachment).

SpFR provides packet transfer in two modes: SpaceFibre mode and GigaSpaceWire mode and support both SpaceFibre and GigaSpaceWire network interfaces. Both have the same PHY layer. SpaceFibre or GigaSpaceWire communication mode is selected by the operation mode setting. The transmission rate of each port is up to 3,125 Gbps.

c) The licensed by ELVEES two SpaceFibre IP cores has the following general features:

- Compliant with the SpaceFibre standard. Supports all the standard functionality
- Optimized for low latency operation





- Highly configurable, giving flexibility through generics in the VHDL source. The following characteristics can be configured:
- Loopback mode
- Receive elastic buffer size
- Number of Virtual Channels (from 2 up to 256)
- Size of the Virtual Channel buffers
- Size of the retry buffers
- The Quality of Service parameters can be configured in real time during operation
- Simple data and broadcast interfaces based on standard input and output FIFO interfaces
- Possibility to start one end of the link in a low-power mode waiting for the other end to become active
- Status and error reporting
- Data and broadcast babbling idiot protection

It is planned to provide an opportunity to work in SpaceFibre network via space qualified fiber-optic transceivers on the ELVEES silicon solutions.

The Standard Cell Library developed by ELVEES for the MCDE50F microprocessor is designed for the 90nm CMOS Logic. The library supports most commonly used basic Boolean functions with multiple drive strengths. While satisfying the performance and power requirements, it was optimized for area efficiency.

The library contains the latch positive-edge and negative-edge version of the clock gating cells.

Power dissipation is becoming an increasingly bigger challenge in chip design as the integration level of circuits becomes large while clock frequencies also continue to increase. ELVEES provides clock-gating cells to help automate the process of power dissipation management. The library contains tristate cells.

The SpaceFibre IP-core PHY Receiver area is 3.125 Gbps - 0.37 sq. mm.

The SpaceFibre IP-core PHY Transmitter area is 3.125 Gbps - 0.37 sq. mm.

Two variants of the transmitter are developed: with VML and with CML output driver.

V. VIDEO ANALYTICS AND COMPUTER VISION CAPABILITIES FOR MCDE50F

Applications of video analytics and computer vision today are extended to new markets. And they become relevant for aerospace applications as well. Adapting applications for a new platform can be difficult. Open standards help to unify the code for any platform.

One of such standards is Khronos OpenVX.

The standard includes more than 40 functions of computer vision with graph representation. The standard allows to implement the functions of computer vision, using hardware accelerators or programmable processors. The OpenVX standard also includes the Neural Network Extension. The

OpenVX guarantees the same functionality on any platform on which it is implemented and supported.

The standard defines a set of regular data structures: tensors, images, arrays of values (coordinates, structures with points of interest). Such data is easier and more efficient to process on a specialized semantic processor. The built-in DMA allows data to be loaded into the internal memory asynchronously: while one patch of data is being processed, the DMA transfers a new one. The CPU forms a chain of tiles for processing and starts a calculation. This procedure continues as long as untreated tiles remain. A multi-buffered calculation scheme is used. Thus, the total computation time t is the maximum time of the t_{hw} processor and the time of the copy t_{mem} :

$$t = max(t_{hw}, t_{mem}).$$

Neural networks computation are based on four algorithms:

- convolution layer computation;
- non-linear activation function;
- pooling layer;
- fully connected layer.

The main time convolution layers demand (up to 90%).

Input image for convolution layers is organized as a set of multi-channel elements and is characterized by depth, width and height. The calculated value of each channel is the sum of weighted values of the channels with filter kernel

 $LP'(i, j, c) = \sum_{m}^{K_{H}} \sum_{n}^{K_{W}} \sum_{k}^{D_{0}} W(c, m, n, k) \cdot P(i + m, j + n, k)$ $i = 0 \dots (H - 1), j = 0 \dots (W - 1), c = 0 \dots D_{1}$

This can be represented as matrix multiply

$$P'(n,c) = \sum_{k}^{K_{H} \cap W} P(n,k) \cdot W(k,c)$$

n = 0 ... (H · W). c = 0 ... D₁



Fig. 4. Data layout representation for matrix multiplication. Pij - vectors of initial elements from D0 values; Wmn - vectors of weights from D0 values; P'in - vectors of the result elements from D1 values.

Data in the image matrix is duplicated. Except for the rows corresponding to the elements near the boundary, the same vector of the original element occurs. $K_H \cdot K_W$ times. This way of organizing data can significantly reduce data transfer overhead and allows to calculate tensor to matrix conversion on the fly (as shown on Figure 4). Effective support for the matrix

multiplication operation is the key to calculate the convolutional layer quickly.

The figure 5 shows a comparison of the performance for the MCDE50F processor with an Intel processor (i5-2400 3.4 GHz). For the 100% reference point the Intel platform was taken. Testing was performed on images with a resolution of 1920x1080 pixels.



Fig.5. Comparison of the computing performance on OpenVX functions fpr MCDE50F and Intel processors.

Processor platform	VGG16, fps	AlexNet, FPS
MCDE50F	2.7	23.1
Intel i5-2400	1.6	9

Fig. 6. Comparison of the computing performance on Neural Network topology for MCDE50F and Intel processors.

Effective use of DMA, the optimal balance of computation and data allow to achieve better results in comparison with the desktop Intel CPU platform (Fig.6).

For E50 (ELVEES video Engine) ELVEES provides an effective CC ++ compiler and a set of optimized libraries, making it easy to port algorithms to the new platform. OpenVX library provides optimized assembler implementation of all functions, as well as extra functions commonly used in vision applications.

VI. CONCLUSION

- 1) The Video and Image microprocessor MCDE50F with SpaceWire/GigaSpaceWire and SpaceFibre links is designed for a wide range of space applications, image and video processing in space "smart" cameras with recognition capabilities and deep learning in satellites and spacecraft systems, as well as in space robotics
- 2) MCDE50F microprocessor provides ability to perform most of today's video analytics algorithms for two or more FullHD 1080p streams at 30 frames per second rate with a peak consumption of less than 5 Watt. It is achieved with universal programmable cores and CNN accelerators that provide the optimum ratio between the area, power and performance.

REFERENCES

[1]. Next Generation Processor for On-board Payload Data Processing Application ESA Round Table Synthesis, ESA, TEC-EDP/2007.35/RT, October 2007

[2]. S.M. Parkes, C. McClements, M. Dunstan and M. Suess, "SpaceFibre: Gbit/s Links For Use On board Spacecraft", International Astronautically Congress, Daejeon, Korea, 2009, paper IAC-09-B2.5.8

[3] "D2.1 - SpaceWire-RT Outline Specification", SPACEWIRE-RT Consortium, 06.09.2012.

[4] "D5.1 - SpaceWire-RT ASIC Implementation Feasibility Summary Report", SPACEWIRE-RT Consortium, 09.05.2013 [5]Sheynin Y., Olenev V., Lavrovskaya I., Korobkov I., Dymov D. STP-ISS Transport Protocol for Spacecraft On-board Networks. // Proceedings of 6th International Conference SpaceWire 2014, Athens, Greece, 2014 ISBN: 978-0-9557196-7-7 pp. 26-31

[6] Olenev V., Lavrovskaya I., Sheynin Y., Korobrov I., Suvorova E., Podgornova A., Dymov D., Kochura S. STP-ISS transport protocol for SpaceWire on-board networks: development and evolution. International journal of embedded and real-time communication systems. 2014, v/5(4), October-December, pp.45-76.

Optical Switches with No Moving Parts for Space Applications

Components, Short Paper

David Poudereux, Juan Barbero Alter Technology TÜV Nord Tres Cantos, Madrid, Spain David.poudereux@altertechnology.com

José Manuel G. Tijero, Ignacio Esquivias Universidad Politécnica de Madrid Madrid, Spain Iain Mackencie ESTEC European Space Agency Noordwijk, The Netherlands

Abstract: We present the work done under the ESA Invitation-To-Tender (ITT) "Optical Switches with No Moving Parts for Space Applications". In this study the consortium reviewed the technical requirements of the space applications envisioned for solid state optical switches. After the technology selection a tradeoff was performed to select the final optical switches, based on three technologies (Magneto-Optic, Bulk Electro-Optic and Waveguide Electro-Optic) and fabricated by four different manufacturers. A test plan was developed, consisting of Thermal Vacuum Cycles, Mechanical Tests (vibration and shocks) and Radiation Test (gamma radiation). The results of the tests indicated that the three technologies are able to be used in space environment, that some technologies are more appropriate for some applications, and that some failure problems should be fixed in specific devices.

Index Terms— Relevant indexing terms: SpaceWire, Solid State Optical Switches, Spacecraft Electronics.

I. INTRODUCTION

Optical switching is a generic building block for many optical systems and it has been proposed for a wide variety of future space applications, i.e., simple redundancy, providing fast isolation, or low speed modulation of the optical intensity.

The use of solid state switching greatly improves the reliability of optical switch technology when compared with the use of bulky mechanical switching systems [1]. In addition to this benefit solid state switching can provide faster switching speeds [2] which is required in SpaceFiber and in some optical switching applications.

Commercial off-the-shelf solid state optical switching technology exists in many forms developed for terrestrial markets such as Telecom or optical sensing; however, a comprehensive analysis of these technologies has not been previously performed with the aim of examining the challenges and benefits of adapting them to space applications.

It is an objective of the ESA to examine the suitability of solid state fiber optic switches to meet future space applications and define a complete technology development and space qualification roadmap for the most suitable solid state optical switch technology to be used in satellite payloads.

This paper summarizes the work done by the authors in response to the ESA ITT "Optical Switches with No Moving Parts for Space Applications" aimed at making a preliminary study of the maturity of the technology and defining a start point for the space qualification roadmap of these devices. The paper is organized as follows: Section II is devoted to the identification of promising space applications and the requirements imposed to the optical switches by these applications. The conclusions of an analysis of the state-of-theart at market level of several optical switch technologies as well as the trade-off analysis for selecting the technologies and the devices to be tested are reported in section III. The central part of the work is the analysis of the results of the mechanical, thermal vacuum and radiation tests which is made in section IV. Finally, the conclusions and recommendations are summarized in section V

II. REQUIREMENTS OF THE SPACE APPLICATIONS

Several space related applications of the optical switches were identified. The envisaged applications are the following:

- 1- CO₂ Monitoring Lidar
- 2A- Atom Sensor-A (750nm)
- 2B- Atom Sensor-B (1580nm)
- 3- Optical Sensing
- 4- Digital Communications
- 5- Local Oscillator-Distribution
- 6A- Optical Communic. T. (5 KWpeak, 1s)
- 6B- Optical Communic. T. (10 W, 100 ms)
- 6C- Optical Communic. T. (100 mW, 500 ns)
- 7- Optopyrotechnics
- 8- Laser Interferometry

To review the requirements of the applications, the consortium contacted with experts in the applications, each application was analyzed, and eleven sets of requirements were defined. These requirements are summarized in Table I.

TABLE I. SUMMARY OF REQUIREMENTS (NLF: NON-LIMITING FACTOR)

APP. CODE	#1	#2A	#2B	#3	#4	#5	#6A	#6B	#6C	#7	#8
Wavelength (nm)	~2000 and 1550- 1650	767/780	1534- 1560	1520- 1570 nm	850 nm	1525- 1565	1550 1064	1550 1064	1550 1064	980+/-15	1064
Number of Inputs	2 (typical); 8 maximum	1	2(4)	1	8	1 (better 4)	1 (better more)	1(better more)	1(better more)	2	2
Number of Outputs	1	1	1	4	8	4	2 (better more)	2(better more)	2(better more)	8 (better 40)	1
Max. Input Power (mW)	≥ 50	≥ 300	≥ 1000	≥10 mW	≥ 1	≥ 40	≥ 10000 average ≥ 5 KW peak	≥ 10000 average	≥ 100	≥ 7500 peak	≥ 3000
Fibre Type	PMF	PMF	PMF	SMF	MMF (50/125)	PMF	PMF SMF	PMF SMF	PMF SMF	MMF (105/125)	PMF
Switch Speed (µs)	< 10	< 0.3	NLF.	< 50 µs	< 5	Irrelevant	Irrelevant	Irrelevant	< 0.5	Irrelevant	NLF
Number of Switch Cycles	$> 5 \cdot 10^9$	very high	NLF	5 billion	> 100	>1000	>1000	>1E8	no wear out	1000	NLF
Lifetime (years)	>3 years LEO	15 years GEO	15 years GEO	15 years GEO	15 years GEO	15 years GEO	15 years GEO	15 years GEO	15 years GEO	15 years GEO	5 years 1 AU
Crosstalk (dB)	> 30	NLF	NLF	> 40 dB	> 30	> 30	> 20	> 20	> 20	> 50	NLF
Insertion Losses (dB)	< 3	< 1	< 1	< 2 dB	< 2	< 1	< 1.5	< 1.5	< 3	< 1	<1

III. ANALYSIS OF TECHNOLOGIES AND DEVICE SELECTION

The potential of different switch technologies for accomplishing the selected requirements was examined. The operating principles, advantages and inconveniences of seven different technologies were analyzed: Bulk Electrooptic (B-EO), Waveguide Electro-optic (WG-EO), Magnetooptic (MO), Acousto-optic (AO), Liquid Crystal (LC) and Thermo-optic (TO).

After the comparison between the requirements in table I and the specifications of commercial products the main conclusions were the following:

• There are no commercial products using TO technologies. In consequence this technology was not further analyzed.

• LC technology can be only tested through custom components, as it is not directly available. No manufacturer showed interest in cooperating in the project. In consequence this technology was not further analyzed.

• All applications require as minimum as possible insertion losses, in any case < 3 dB and in some cases < 1 dB. The products based on AO present IL > 3 dB. This technology is complex regarding the electronic driving, and its advantage is a high switching speed which is not required by any application. In consequence it was not further considered.

• In conclusion three technologies were considered: WG-EO, B-EO and MO. Within these technologies there are different manufacturers supplying optical switches with different characteristics (wavelength, maximum power, type of fiber, etc) and using different materials. Some of them were selected after a tradeoff process. Two types of tradeoffs were considered:

• Tradeoff at Technology Level: The applicability of each technology was analyzed considering the space related possible applications without considering a particular device.

• Tradeoff at Component Level: In this case each part type was analyzed considering also the manufacturer. The output of this analysis was the final selection of components for the testing activities summarized in Table II, where the nominal characteristics of each device and the number of components are included.

In summary, eight different commercial models of optical switches fabricated by four manufactures using the three technologies (MO, B-EO and WG-EO) were selected for the testing.

TABLE II. SELECTED OPTICAL SWITCHES

Manufac.	Туре	Max. Power (W)	Fiber Type	Sw. Speed (µs)	CT (dB)	IL (dB)	No
	МО	0.3	SMF	5-200	50	0.8	7
Agiltron		0.3	PMF	5-200	50	0.8	7
(US) B- EO		5	SMF	5-200	50	1	1
	B- EO	0.3	SMF	< 0.3	25	0.8	7
Epiphotonics (US)	WG- EO	0.001	SMF	< 0.01	30	5	7
BATi (US)	B- EO	0.5	SMF	0.06	>20	<1.5	10
Primanex (CH)	мо	0.5	SMF	200- 400	>40	< 1.3	1
		5	PMF	200- 400	2 10	1.8	7

IV. TEST RESULTS

After procuring the selected components, a test plan following the usual test procedures for space applications was carried out.

The test plan consisted in the following steps and items:

- Initial Electro-Optical characterization
 - Insertion Loss (IL)
 - -Crosstalk (CT)
 - **Response** Time
 - Polarization Dependent Losses/Polarization Extinction Ratio (PDL/PER)
- Mechanical Test
 - Vibration -
 - Shock test
- Thermal Vacuum Test
- Radiation Test
- **Destructive Physical Analysis**

All the samples were electro-optically characterized and the results were compared with the nominal specifications in the manufacturer datasheets. The measured values were in some cases better and in other worse than the nominal values. They were taken as a reference

A. Vibration Test

A 10 cm side vibration cube was used for fixing the samples. This cube is free of resonances up to more than 2000 Hz and it can be rotated to allow changing the axis easily. The levels of the vibration are summarized in Table III.

The duration on each of the 3 orthogonal axes test was 3 minutes. One sample of each part type was monitored during test, and the output of the switches was changed alternatively with a periodicity of approximately 10 seconds. Each switch had three optical fibers (one input and two outputs) and two electric wires (except EpiPhotonics switch which needs 24 electric wires to control the applied voltages). Before the random vibration a pre-vibration test was performed in each axis looking for possible undesirable resonances. Two accelerometers collected the data, one uniaxial, fixed to the shaker and a second triaxial, fixed to the cube. Each one provides different information, but both data are correlated. The data taken for the accelerometers attached to the shaker and cube of the first run in the X axis are shown in Figure 1.

Figure 2 shows the stability of the measured optical power at each output of the switches during the vibration test. There are in fact two sets of data for each output, one for the values in the maximum (ON state) and other for minimum (OFF state). The IL and the CT of each output are calculated by simultaneously measuring the power at both outputs. The flatness of these plots reveals that the optical power, i.e. the insertion loss, kept constant during the vibration test.

TABLE III.	VIBRATION]	LEVELS
------------	-------------	--------

Frequency (Hz)	Protoflight Level
20	0.52 g ² /Hz
20-50	+6 dB/Octave
80-800	0.32 g ² /Hz
800-2000	-6 dB/Octave
2000	0.052 g ² /Hz
Overall	20 grms



Fig. 1. Random vibration in X axis.





B. SRS Shock Test

The 500g half Sine Pulse 1ms duration shock profile was converted to a SRS shock test. Three impacts were made in each axis. All the electro-optic parameters of all the switches were measured after the test showing no variation. Only one of the switches (EpiPhotonics) was damaged during this test. One of its arms (the input one) was unglued.

C. Thermal Vacuum Test

The thermal cycling tests were done under the conditions detailed in Table IV.

Five thermocouples were fixed to the hot/cold plate to control and measure the temperature. Two runs were performed at the beginning in order to test the same part types of all the switches that underwent the vibration test. The monitoring set-up was very similar, the only difference being that the drivers and the splitter were placed inside the vacuum chamber, isolated from the cold/hot plate. A previous validation test was performed to ensure the proper functioning of the photodiodes.

TABLE IV. VACUUM THERMAL CYCLING CONDITIONS

Vacuum Thermal Cycling Test Proposed					
Tmin	-10° C	-5°C			
Tmax	75°C	+70°C			
Dwell time at Tmin & Tmax	2 hours	2 hours			
Pressure	<10 ⁻⁵ mbar	<10 ⁻⁵ mbar			
Temp rate	<5°C/min	<5°C/min			
Number of cycles	1	7			



Figure 3. Evolution of the optical power along the radiation test

D. Radiation Test

The Gamma Radiation campaign was performed at the facilities of the National Center of Accelerators (CAN) in Seville (Spain). The dose rate was ~210 rad/(Si)h with an accumulated dose of ~100 krad(Si) and a duration of ~480 hours. The switches were placed in an aluminum box covered with PMMA to achieve the condition of electronic balance. The optical output power was monitored along the whole radiation test.

The evolution of the optical power through all the outputs (two per switch) in both states ON/OFF (output 1/ output 2) is plotted in Figure 3. The interruptions of the radiation, inherent to the regular operation of the facility, are also highlighted with black marks. These interruptions lasted less than 2 hours. Note that the output optical power is sensitive to the radiation interruption, mainly in the OFF states.

E. Destructive Physical Analysis

After the final electro-optical characterization, a destructive physical analysis was made to some optical switches. Only six part-types were analyzed since two part-types differ only in the fiber type (SM or PM) (see table II).

Only one switch (EpiPhotonic) showed damage. It was the switch which input fiber was broken in the shock test. It seems that the crystal spliced to the fiber was unglued in the test. Likely, this failure is not inherent to the WG-EO technology since the aligning of the crystal in the photonic circuit is not more complex than for other technologies.

F. Evaluation of the test results

The results of the electro-optical characterization after each test were compared with the corresponding results for each parameter prior to the tests. The averaged standard deviation of all the measurements of each reference part type has been taken as a value of the uncertainty and the deviation of the values of each parameter after each test has been compared with this uncertainty. The Student's t-distribution was used for this statistic. Judicious criteria were defined to decide whether a part type passed the test. According with these criteria all the tested samples, except the mentioned EpiPhotonic switch, passed all the tests.

V. CONCLUSIONS AND RECOMMENDATIONS

The conclusions of our study are the following:

• B-EO and MO technologies are excellent candidates for the space applications analyzed. They respond very well under typical space conditions as radiation, vibration, shocks and thermal vacuum.

• B-EO technology behaves slightly better than MO technology and it is the choice for those applications requiring high switching speed. However, its crosstalk is worse

• MO technology exhibits very good properties, except for the switching time. Moreover, there are many manufacturers and commercial products fabricated with this technology.

• WG-EO technology is very fast. Its mechanical problems could be solved if the gluing of the crystals to the socket is improved to resist the shock tests. However, the switches are very complex to handle and control, their insertion losses are very high and the cross talk very low. The high-speed requirement of some applications is covered by B-EO technology and therefore WG-EO is not recommended for space applications.

• Switches are more sensitive to temperature than to radiation and vibration.

• No difference has been found between the behavior of polarization maintaining fibers and single mode fibers.

• The low power (single mode fiber) MO devices and the high power (multimode fiber) devices from the same manufacturer (Agiltron) behave alike under the tests when operated at low power.

Based on these conclusions the recommendations were:

• To use B-EO and MO technologies for the space applications analyzed. B-EO should be used for applications requiring high switching speed while MO technology should be used for those applications requiring a minimum crosstalk.

• To keep constant the device temperature if constant output power is required.

REFERENCES

- [1] S. Abad, F.M. Araujo, L.A. Ferreira, F. Pedersen, M.A. Esteban, I. Mckenzie, N. Karafolas "Applications of FBG sensors on telecom satellites," Intl. Conf. on Space Optics ICSO'2010, Rhodes, Greece, October 2010.
- [2] N. Venet, M. Sotom, H. Gachon, V. Foucal, M. Pez, V. Heikkinen, ... & S. Pantoja, "High-throughput optical interboard interconnects for next-generation on-board digital transparent processors," In International Conference on Space Optics, vol. 7, p. 10, October 2014.

SERDES Single Event Effects in 65nm Flash-Based RTG4 FPGA

PAPER NOT PUBLISHED

Thursday 17 May

On-board Equipment & Software (Long)

SpaceFibre based Mass Memory

Extreme Rapid Mass Memory Unit (ExtRa MMU) enabled by SpaceFibre Session: On-board Equipment & Software

Richard Wiest, Ottmar Ried, Paul Rastetter Space Equipment Airbus Defence and Space GmbH 81663 Munich, Germany <u>richard.wiest@airbus.com</u>

Abstract

The demand in data storage capacity as well as data rate is steadily increasing in ENS (Earth Observation, Navigation and Science) satellite missions. This requires a mass memory unit (MMU), a product providing an on-board data acquisition, storage and playback capability between payload and downlink transponder of a satellite.

Associated huge data rates require high speed serial links (HSSL) combined with a sophisticated protocol. For the next generation of Airbus Defence and Space mass memory products the SpaceFibre protocol has been selected as most suitable protocol.

SpaceFibre is used for internal and external communication, implemented in latest FPGA technology and connected through copper or potentially optical.

Index Terms

Mass Memory Unit (MMU), Payload Data Handling Unit (PDHU), Flash, SpaceWire, SpaceFibre, High Speed Serial Link (HSSL), ExtRa (Extreme Rapid)

I. INTRODUCTION

Since the 90s a rising demand for on-satellite data storage can be observed, which usually cannot be satisfied with space grade available components. In the early phases those capacities were in the range of some Gbit, whilst today we speak about dozens of TBit, coming together also with a huge increase in data rates, reaching nowadays values of up to tens of Gbps.

Such progressive memory capacities have always been realized accommodating commercially available memory components, like SDRAM, DDR and modern Flash Memories, together with necessary mitigation measures to make them useable and reliable in the Space context. Consequently Airbus established a complete portfolio of products for different use cases, mainlyin the context of ENS missions, with e.g. optical or radar instruments producing significant amounts of data. Erich Weih, Michael Stähle, Günther Lohse, Martin Steiner Space Equipment Airbus Defence and Space GmbH 88090 Immenstaad, Germany erich.weih@airbus.com



French Riviera imaged by Sentinel 2 MSI on 27 June 2015 Fig. 1. Sentinel 2 MMU

As new key enabling technologies have been recently made available to the space community, like high-speed serial links and sophisticated protocols, complex FPGAs as well as newer generation memory technologies the whole product portfolio within Airbus is undergoing a complete overhaul.

Airbus is targeting unique performance characteristics to meet current and upcoming customer requirements. For the realization of high speed serial links (HSSLs) a trade-off has been performed on the best implementation, leading to baselining SpaceFibre as communication protocol to be used on all, i.e. internal and external, HSSLs.

II. MOTIVATION FOR A NEW MASS MEMORY PORTFOLIO

State-of-the-art high end (high speed and high capacity) MMUs implement today a capacity of roughly up to 16 TBit with an aggregate data rate of up to 16 Gbps. An example of such an unit is the NISAR Solid State Recorder (SSR for customer JPL/US) shown in picture below.
III. NEXT GENERATION MASS MEMORY



Fig. 2. NISAR Solid State Recorder

Key Performance Indicators of the current Airbus Defence and Space portfolio are:

- Aggregate data rate up to 16Gbps, Storage capacity up to 16Tbit (EOL)
- Flexible real-time SW based embedded File Management System with PUS services and CFDP
- Simultaneous record and replay
- CCSDS conform output Data Formatting at a data rate of up to 4.0 Gbps (net)

In the recent years, new emerging technologies have appeared for space applications which allow thinking in higher performance classes. The most important technologies for MMUs, which allow boosting the system performance, are

- 1. High-density and high-speed NAND Flash memory devices which improve the Data Storage Function in terms of storage capacity and data rate.
- 2. High-speed serial links (utilizing the benefits of the SpaceFibre standard as VC, QoS) which improve the Communication Network bandwidth and reliability.
- High-performance space-grade FPGAs (e.g. Microsemi RTG4 or NanoXplore BRAVE-Large) which allow for higher processing performance, higher integration density of complex functions and high speed serial links (HSSL) allowing direct SpaceFibre protocol implementation.

The efficient combination of these emerging technologies with adequate data processing and control algorithms are the key items for the next generation mass memory architecture. This architecture is fully based on a SpaceFibre network architecture, enabling the required data throughput whilst implementing necessary routing and failure correction mechanisms.

This completely new product generation will allow about four times the performance of state-of-the-art equipment, i.e. up to 64 Tbit capacity and an aggregate (record/replay) data rate of up to 64 Gbps. The generic functional architecture of a Solid State Mass Memory can be depicted as such:



Fig. 3. Generic Mass Memory Architecture

The key elements listed are:

- The Memory System Management Function (MSM)
- Memory Power Control Function (MPC)
- Data Storage Function (DSF)
- Data Input Processing Function (DIP)
- Data Output Processing Functions (DOP)
- Data Processing Functions Frontend, Background and Backend Processing (FEP, BEP, BAP)
- Communication Network
- Power Control Network
- Descentralised Power Distribution

The physical implementation of those functions can be done in three different kinds, depending on number/type of interfaces, total capacity, operational environment and requested complexity/autonomy:

- 1. **Compact architecture**, i.e. implementation of all functions listed above into one single module, offering a highly integrated and efficient solution, but with limited capacity and data rate performance.
- 2. Multi-module solutions, either with a
 - a. **Sliced architecture**, i.e. multiple instances of identical modules, implementing the necessary minimum of functions
 - b. **Centralized architecture**, implementing all functions on different modules, with multiple modules of same type to meet capacity, data rate or processing demands. Various network architectures are possible to best meet the mission requirements.

A. Physical architecture

The generic architecture can be mapped to a physical implementation as sketched in the following figure, which shows the so called Array Architecture as an example for a *centralized architecture*.



Fig. 4. Example Array Architecture

Mapping of the mass memory functions to hardware modules depends on complexity, redundancy requirements and FPGA resources to implement the logic.

The Memory System Management Function is realized by a Memory System Supervisor Module (MSS), which is currently based on an LEON2-FT processor. It runs the application software; the file management system and low level drivers to control the hardware based mass memory functions and interfaces. This function includes also the external TC/TM interfaces for communication with the On-Board Computer (OBC). With power on of the Power Converter Module (PCM) the related MSS is powered, too, and starts operation of the system. The PCM converts the primary supply voltage to secondary voltages, which are distributed to each single internal module of the data processing path.

The MSS powers and configures the other modules according to the operational needs. In the figure above the blue boxes are powered and the white ones represent cold redundant modules.

In the most cases Data Input and Front End Processing Functions are mapped to the IO Module (IOM), which performs the actual high speed communication with the instruments and downlink system.

The Flash Memory Module (FMM) represents a core element and includes the Data Storage Function. The storage capacity on one FMM is about 8 Tbit at an IO performance of 8 Gbps.

A typical data processing function is compression. In this example compression is performed as a Background Processing Function using a dedicated Compression Module (COM).

The anticipated performance of the given example is 16 Gbps input data reception and storage, up to 4 Gbps background data compression and up to 10 Gbps downlink data rate which is a typical value for modern and next emerging downlink systems as Relay Satellites and Laser Link technologies.

The PCM and MSS are redundant and can be independently powered. The data processing and storage modules are normally n+1 redundancy configurations.

B. Interface/Protocol

The emerging SpaceFibre network protocol is a preferred candidate for implementation of the serial high speed communication network for the following reason: SpaceFibre will become an ECSS standard and fulfills some important criteria which are reflected in the following:

- High data transfer rate
- Error detection/handling
- Bi-directional interface (full-duplex)
- Small FPGA implementation cost
- Interfacing possible with available low-end solution
- Virtual channel layer with Quality of Service support
- Possibility to have access with 'commercial' available test equipment

The SpaceFibre protocol has no performance limitations, thus a gross performance of up to 6.25 Gbps is presently favored and fits well to the performance class of the mass memory functions. The SpaceFibre protocol introduces some handling and header overhead in the data transfer, 4.5% in unidirectional and 7% in bi-directional mode; but this is a common value for the bi-directional functionality. The biggest reduction (20%) is based on the 8B/10B coding scheme used for the serialization, therefore a future adaptation of the standard towards higher efficient coding schemes (e.g. 128B/130B) would be appreciated. However this coding

scheme is commonly used for the intended speed grade and allows the interconnection of different communication devices provided by various manufacturers.

Error detection/handling is used to achieve a reliable data transport. SpaceFibre detects errors and handles corrupted data units, here called frames, by autonomous repetition of the affected data units in case of errors. No user interaction is necessary.

SpaceFibre is a bi-directional interface and enables also full-duplex communication. This is a great advantage for the system design and allows more versatile data communication scenarios.

SpaceFibre has been designed to have a small FPGA footprint and implementation costs. This eases the implementation of multiple SpaceFibre instances and imposes fewer constraints on the FPGA design; more FPGA resources for other functions are available. FPGAs with already hard-coded SERDES functionality are usable (with or without 8B/10B coding) as well as in conjunction with external SERDES devices; which offers additional failure isolation. Furthermore with external SERDES devices even older FPGA technologies such as the RTAX2000 can be interconnected to a SpaceFibre network.

SpaceFibre implements the concept of virtual channel layers with Quality of Service support. This is a powerful method to run independent data streams using only one physical interface. Each virtual channel is acting as an individual interface and the QoS methods allow the management of priority, scheduling, and bandwidth per virtual channel. For a mass memory system especially the separation of high speed user data and high-priority command and monitoring is needed on a common SpaceFibre network.

For SpaceFibre 'commercial' test equipment is available and already in-use. It allows different levels of link testing, link monitoring, error injection and localization, performance measurement and protocol verification.

SpaceFibre is fully downwards compatible with SpaceWire data packets. This enables a performance upgrade without general architectural changes; multiple instrument data streams can be easily inserted within SpaceFibre links.

C. Physical layer

SpaceFibre is designed to support both physical interface types – copper and optical . In principle no changes are necessary to adapt the SpaceFibre interface from copper to optical interconnect. If the SpaceFibre specification is followed and the optical transceiver has the CML interface definition (which is the standard) then the optical transceiver could be used in front of the copper interface.

Copper is widely used and well known for the implementation requirements and for the drawbacks, as there are mainly the increased harness mass, no galvanic isolation, and bounded bandwidth.

On the other hand fibre optical links offer improvement to the mentioned drawbacks of copper interconnects. Whilst handling of optical cables and interconnects needs careful consideration, commercial solutions are already well introduced in the market (especially in telecommunication).

As a bottom line SpaceFibre supports both copper and fibre interface with one specification and protocol.

For equipment internal interfaces copper links have been selected, as an optical implementation will complicate electronics and backplane design (size, mass, cost). For the external interfaces to up/downstream equipment flexibility is offered to either implement optical and/or copper interfaces.

D. Data-flow

In general the MMU manages the internal data flows from instrument interfaces to memory and from memory to downlink interfaces. In between data processing functions can be placed for compression, encryption, quick look, data analysis or other tasks. The general flow is indicated in Fig. 3, but indeed the internal physical communication network uses bi-directional SpaceFibre interfaces and allows an optimized physical network implementation independent from the data flow.

Due to the bi-directional nature of the SpaceFibre links it becomes obvious that input and output interface functions should be combined to a common IO function. This is also supported by the upcoming FPGA technologies which provide much more logic resources to implement both functions.

Having an IO Module the next step is not far away to interconnect the MMU communication partners via a common network, which is also based on the SpaceFibre standard. Similar solutions are already realized, as e.g. for Bepi-Colombo or probably in future for JUICE with SpaceWire links and SpaceWire routers.

From this follows that the MMU might provide a dedicated SpaceFibre router, which can be interfaced by OBC, instruments, downlink system and the MMU itself. Due to the virtual channels control and high speed data can be exchanged in any direction using the same network. This means the OBC can control all network nodes and the MMU can receive highspeed data from the instruments for data recording and can transmit high-speed data to the downlink system for data replay using the same interfaces and router based network.

A principle network configuration is shown in Fig. 4.

E. SpaceFibre FPGA Implementation

The high transmission rates of SpaceFibre makes an implementation with fabric logic and general purpose IO impossible. Therefore dedicated silicon solutions are necessary to implement HSSL, either by external solutions with parallel interface or by specialized hard IPs which are available in newer FPGA technologies.

FPGA External HSSL solutions (e.g. TLK2711) are of interest in two main ways. First, they give the possibility to use mature FPGA technologies (e.g. RTAX2000) without HSSL IPs for SpaceFibre implementations. The second aspect is the

decoupling of an external interface from the main FPGA. This is important because FPGA pins must not be exposed to an external connector due to EMC rules. An external solution consumes much more board space, user IO's and has higher power consumption as well as a challenging board design. The FPGA resource utilization remains basically the same as for internal HSSL solutions.

FPGA Internal HSSL hard IP solutions simplify the board design significantly to two differential line pairs per lane. Unfortunately, at the moment is no qualified buffering solution available which therefore limits the usage to unit internal connections.

All HSSL technologies, either internal or external follow basically the same scheme of a parallel interface for the TX and RX path with only minor differences like 8b/10b encoder available or not. This similar interface simplifies also the portability between different technologies. Nevertheless internal communication systems must also be able to handle the high bandwidth of SpaceFibre. Which means for example, a 32bit parallel data stream at a clock frequency of higher then 80 MHz is necessary to transport the data of a single 3Gbit lane. Routing and processing of data at this speed must be achieved to keep the overall performance.

The radiation hardened RTG4 FPGA from Microsemi is at the moment the most promising candidate for implementing HSSL systems for space. It covers a total of 24 bidirectional lanes each capable of 3.125 Gbps data rate. The resource consumption of less than 3% of the chip for a single lane SpaceFibre implementation allows designs with a high number interfaces. The European BRAVE-Large FPGA development is a further future possibility with integrated HSSL hard IP. The specified data rate of up to 6.25 Gbps even surpasses the RTG4 capability in this regard. Resource utilization of SpaceFibre implementations are expected at the same level as for RTG4. Both FPGA's have the potential to drive high performance products based on SpaceFibre communication.

As the RTG4 technology is already available in spacegrade on the market it has been baselined for the ExtRa-MMU design, but the RAVE-Large will also be implemented as an option.

F. Test Environment

The backbone of Airbus Defence and Space's Electrical Ground Support Equipment (EGSE) is the Universal Data Management System (UDMS) tool suite (see Fig. 5). It is compatible with the Packet Utilization Standard and supports:

- Data rate assessment
- Various data displays (Plots)
- Synoptics displays for example Bad Block management
- Test automation over scripting interface
- Offline Data Evaluation



Fig. 5. UDMS Test Environment

The SpaceFibre interface fulfills all prerequisites to be integrated into this UDMS environment by a new Frontend Application as depicted in the orange shaded circles above.

This Frontend encloses the project and hardware specific adaptations to the EGSE and to the Unit Under Test (UUT).

SpaceFibre specific commercial off-the-shelf (COTS) test equipment is available on the market such as the STAR SpaceFibre box that can complement the EGSE test setup with UDMS.

The SpaceFibre protocol layers facilitate the fault localization in the system and the verification of the high speed data link through dedicated status information and error messages that can be evaluated during test and integration of the design.

Additional advantages of the SpaceFibre interface for test and integration are:

- Error handling
- Scalable protocol
- Quality of service

G. Software

The in-house developed file management software from Airbus Defence and Space has been especially designed to operate the mass memory unit with high end capacity & data rate. In several projects (NGMMA, Euclid and NISAR) approved functionalities demonstrate the capabilities of the file management software to fulfill the high customer needs. Features of the file management software are:

- Configurable Files- and Directory-Structures
- Extendable TM/TC interface
- Packet Utilization Standard (PUS) for standardized and adoptable communication

- Sophisticated data recoverable non-volatile file system
- CCSDS File Delivery Protocol (CFDP)
- FDIR mechanisms

Heritage design and state-of-the-art software architecture allows introducing new technologies easily such as SpaceFibre to achieve high transmission rates.



Using SpaceFibre as internal communication technology (reliable/failure-tolerant) helps to decrease the CPU utilization while transferring mass data within the mass memory unit. The reduction of CPU utilization makes it easier to achieve software performance requirements and opens new capabilities for other mission related operations.

H. Common Memory Module

The complete next generation Airbus memory portfolio is based on the same key building block, namely the CM^2 (Common Memory Module).

The following key elements are embedded in the CM² and are fully qualified on component, assembly and building block level:

- RTG4 FPGA (European BRAVE FPGA in preparation)
- Local power supply
- DDR2 Memory & Controller
- Flash Memory & Controller
- Reed Solomon EDAC for Flash and DDR2
- HSSL (High Speed Serial Link)
- Debug and support interfaces

I. ExtRa-MMU

The Extreme Rapid Mass Memory Unit (ExtRa-MMU) is a high end capacity & high end data rate mass memory unit with a centralized architecture. Scalability is achieved through increasing either the number of memory modules (scaling of capacity), number of I/O modules (data rate & interfaces) as well as processing/compression modules.

Like all other next generation Airbus Defence and Space mass memory products it is based on the CM^2 core building block, which is forming the heart:



Fig. 7. Airbus next generation Mass Memory product portfolio

Key highlight of NG Products are:

- Three architectures: Compact Single Module (SBMM), Multiple modules centralized (ExtRa MMU) or sliced (CORECI2)
- Airbus in-house qualified (2015-2018) next generation Flash device
- Common mass memory core (CM² building block, incl. FPGA/VHDL, Flash, DDR, PoL)
- Data rates and memory capacity are 4 x compared to current products with flexible interfaces
- Features including pre-processing and compression, real-time SW for File Management with PUS services and CFDP, lambda processing, end-to-end coding

The ExtRa-MMU is the most advanced high end capacity & high end data rate mass memory unit with a centralized architecture. Scalability is achieved through increasing either the number of memory modules (scaling of capacity), number of I/O modules (data rate & interfaces) as well as processing/compression modules.

- High Speed High Capacity Mass Memory System, based on a centralized architecture
- Central processor module for Command & Control and file management
- Up to 64 Gbps IO Performance
- Up to 64 Tbit Storage Capacity
- Modular and scalable architecture for interface, data processing, data storage and redundancy concepts
- Compression/Processing as option (on-line or offline)

J. Hardware Status & Test Results

Baseline for all of these new developments are the results achieved in frame of the Next Generation Mass Memory Architecture (NGMMA) study (ESTEC contract), the Space.Server study (DLR contract) and Airbus internal R&D activities. In frame of these studies the different architectures have been consolidated and its performance demonstrated with relevant demonstrators.

The following key elements were demonstrated with these demonstrators:

- Single Board Mass Memory (SBMM) Architecture
- Centralized Mass Memory Architecture
- SpaceWire Router/Concentrator
- SpaceFibre Network incl. SpaceFibre Router
- Common mass memory core (CM² building block, incl. FPGA/VHDL, Flash, DDR)
- Real-time SW for File Management with PUS services and CFDP, end-to-end coding

The modules required for a complete ExtRa-MMU have the following status:

- The flash memory module is in its manufacturing phase. First test results are scheduled for beginning of Q3/18
- The input/output module is in the design phase and is planned to be available end of Q3/18
- Mass Memory Supervisor available from several former programs (TRL9)
- DC/DC Converter available from several former programs (TRL9)
- Mass Memory SW incl. real-time File Management with PUS services and CFDP available
- Mechanics available as based on heritage design
- ExtRa-MMU unit availability by end 2018
- Full qualification of the ExtRa MMU will be completed within 2019.

IV. HIGH SPEED DATA CHAIN

The SpaceFibre interface is perfectly suited to be the backbone of a complete end-to-end high speed data chain (HSDC) onboard a satellite that interconnects all payloads like the mass memory unit, mission instruments, processing units, optical or RF data links.

This SpaceFibre backbone in form of a network comprises one or more interconnected SpaceFibre routing switches. The network architecture can be flexibly mapped to the particular requirements of the mission. In Fig. 6. below an example of a SpaceFibre network is shown that has several routing switches arranged in a ring with, for instance, a four-lane SpaceFibre link between each routing switch. Information can flow around the ring in either direction, giving a total network bandwidth of twice the data rate of the links between routing switches.



Fig. 8. High Speed Data Chain (HSDC)

Such a SpaceFibre network with the aim to remarkably improve on-board data handling and transfer capabilities is developed and validated in the so called Hi-FLY project (www.hi-fly.eu).

V. CONCLUSION

Airbus has a total of more than 40 Payload Data Handling Units (PDHUs) successfully launched and operating in space since 1990 with a further growing institutional and export customer base worldwide including e.g. Japan, Korea, Brazil and US.

All qualified NG technologies (Flash, FPGA, buffer memory, power) will enable very flexible and highest performance products within 2018: CORECI-II, ExtRa MMU and Space.Server. The ExtRa MMU is the flagship in terms of capacity and data rate. This is enabled through implementing all data communication based on SpaceFibre with transmission either electrically or optically.

ACKNOWLEDGMENT

The research leading to these results has received funding from the European Space Agency under ESA contract numbers 22530/09/NL/LvH and 4000122420/17/NL/GLC/fk, from Deutsches Zentrum für Luft und Raumfahrt 50RM1405 and from the European Union's Horizon 2020 Research and Innovation Programme under grant agreement No 776151.

REFERENCES

- [1] NGMMA (ESTEC Contract No. 22530/09/NL/LvH)
- [2] Hi-FLY (European Union's Horizon 2020 Research and Innovation Programme under Grant Agreement No 776151)
- [3] Entwicklung eines EinPlatinen Onboard Speichers mit Netzwerk Funktionalität (EPOS)/Space.Server (Deutsches Zentrum für Luft und Raumfahrt Contract No. 50RM1405)
- [4] GSTP6.2 Extreme Rapid Mass Memory Unit (ESA Contract 4000122420/17/NL/GLC/f

SpaceFibre Camera On-board Equipment and Software, Long Paper

Steve Parkes, Ashish Srivastava School of Science and Engineering University of Dundee Dundee, UK s.m.parkes@dundee.ac.uk Chris McClements, Pete Scott, David Dillon STAR-Dundee Ltd. Dundee, UK

Albert Ferrer Florit, Alberto Gonzalez Villafranca STAR-Barcelona San Cugat, Spain

Abstract—SpaceFibre is a high performance, high availability technology for space flight and other demanding applications. The recent generation of image sensors are capable of data rates of several Gbps. SpaceFibre is ideal as an interface to such an image sensor. STAR-Dundee has designed a complete camera, which incorporates a radiation tolerant FPGA for sensor interfacing and control, and image signal processing. This paper introduces SpaceFibre, the Microsemi RTG4 FPGA, the CMV4000 image sensor and describes the complete SpaceFibre camera.

Index Terms—SpaceFibre, Image Sensor, SpaceWire, Networking, Spacecraft Electronics, Radiation Tolerant, FPGA, RTG4.

I. INTRODUCTION

SpaceFibre [1-4] is a new generation of SpaceWire [5-6] technology which is able to support the very high data-rates required by sensors like SAR and multi-spectral imagers. Data rates of up to 25 Gbps are required to support several sensors currently being planned. In addition, a mass-memory unit requires high performance networking to interconnect many memory modules.

To support the development of spaceflight equipment incorporating SpaceFibre, STAR-Dundee has developed a range of SpaceFibre IP cores and SpaceFibre test and development equipment. The IP cores are being used in radiation tolerant FPGA and ASICs. Recently STAR-Dundee has designed a SpaceFibre camera and a SpaceVPX based processing board to demonstrate the ease with which SpaceFibre can be integrated into spaceflight systems and the substantial advantages it can bring.

This paper first introduces SpaceFibre. It then describes the main components of the SpaceFibre camera; the Microsemi RTG4 radiation tolerant FPGA and the CMOSIS CMV4000 image sensor. The SpaceFibre camera is then described in detail.

II. SPACEFIBRE

SpaceFibre is designed to support high data-rate payloads, including synthetic aperture radar and hyper-spectral optical instruments. It provides robust, long distance communications for launcher applications and supports avionics applications with deterministic delivery constraints through the use of virtual channels. SpaceFibre enables a common on-board network to be used across many different mission applications resulting in cost reduction and design reusability.

SpaceFibre runs over both electrical and fibre-optic media and provides 3.125 Gbps data rate in current radiation tolerant FPGA technology. Higher data rates of 6.25 Gbps data rate per lane are possible with 65nm ASIC technology. SpaceFibre provides a quality of service mechanism which is able to support priority, bandwidth reservation and scheduling. It incorporates fault detection, isolation and recovery (FDIR) capability in the interface hardware. SpaceFibre is designed to be implemented efficiently and has a much smaller footprint than competing technologies such as Serial Rapid IO, taking 3-5% of a Microsemi radiation tolerant RTG4 FPGA [7-8] which allows plenty of room for the application specific logic.

Several SpaceFibre lanes can be operated in parallel (multilaning) to give higher data rates or increased reliability. A multi-lane link can have any number of lanes from 1 to 16. Multi-lane operation provides hot redundancy and graceful degradation in the event of a lane failure, simplifying redundancv approaches maintaining essential and communication services over the remaining operational lanes. When a lane fault does occur, recovery is very fast, taking a few µs. SpaceFibre also supports asymmetric links where some of the lanes can be unidirectional. This is particularly useful for high data-rate instruments where data flow is mainly in one direction, and can save both power and mass.

SpaceFibre is backwards compatible with SpaceWire at the network level, using the same packet format, which allows simple interconnection of existing SpaceWire equipment to a SpaceFibre link or network.

SpaceFibre has a message broadcast capability, which carries eight bytes of user information, together with a broadcast type and channel identifier. This permits, for example, CCSDS unsegmented time information to be broadcast across the spacecraft in a single broadcast message, with low latency. Broadcast messages can be used for time distribution, synchronisation, event signalling, network control and error handling.

The ECSS-E-ST-50-11C SpaceFibre standard will be published in November 2018.

STAR-Dundee has developed a comprehensive set of IP cores for SpaceFibre which are already being used in their first space missions and ASIC designs. Both single-lane and multilane IP cores are available. In addition, STAR-Dundee has a SpaceFibre router design running on radiation tolerant FPGAs. The following IP cores are now available from STAR-Dundee targeted for Microsemi RTG4 and Xilinx FPGAs, and ASIC implementation:

- Single-lane SpaceFibre interface;
- Multi-lane SpaceFibre interface;
- SpaceWire to SpaceFibre bridge;
- SpaceFibre routing switch.

III. MICROSEMI RTG4 FPGA

The RTG4 FPGA is a radiation tolerant FPGA from Microsemi which is fully reprogrammable and which has an integrated flash memory to store configuration information.



Fig. 1. Microsemi RTG4 FPGA Architecture

The RTG4 contains 151,824 logic elements each comprising a 4-input LUT and flip-flop capable of operating at a clock speed of 250 MHz. There are 209 blocks of dual-port SRAM which can each be configured as 512 x 36, 1 kbit x 18, 2 kbit x 9 or 2 kbit x 12 memory blocks. The RTG4 also includes 210 three-port SRAM blocks which can each be configured as 64 x 18, 128 x 12 or 128 x 9. To access external memory there are two high-speed DDR2/DDR3 memory controllers which can operate at 333MHz and support x9, x12, x18 and x36 bus widths. The memory controllers provide optional error detection and correction (EDAC) for the external memory devices.

The RTG4 has 462 math blocks, each of which contains an 18x18-bit multiplier and a 44-bit accumulator. The math blocks can provide 250 MHz pipelined performance giving a total potential DSP performance of 230 GOPS. The math blocks are ideal for many DSP functions including filters and Fast Fourier Transforms (FFTs). A FFT spectrometer, implemented by STAR-Dundee in the RTG4 FPGA, achieved a DSP performance of 100 GOPS [9].

To implement SpaceFibre a high-speed serialiser/deserialiser (SerDes) is required, which takes relatively slow, parallel data and serialises it for transmission at high-speed and de-serialises the received serial data to recover the parallel data stream. The RTG4 FPGA includes 24 high-speed (3.125 Gbps) SerDes on-chip avoiding the need for external SerDes devices. These integrated SerDes make the RTG4 ideal for the implementation of spaceflight data-handling and processing sub-systems with multi-Gbps SpaceFibre interfaces. STAR-Dundee has a range of SpaceFibre IP cores which are optimized for the RTG4. The RTG4 has 16 SpaceWire clock and data recovery circuits, which support SpaceWire interfaces running at up to 180 Mbps.

The 65nm flash process used in the RTG4 devices is intrinsically immune to configuration upsets, and the devices also feature additional radiation protection for data in flip-flops and combinatorial logic elements, embedded SRAM cells and multiply accumulate blocks. The RTG4 is designed to eliminate single-event latch-ups.

IV. IMAGE SENSOR

An image sensor was required for the SpaceFibre camera which had reasonable resolution, was capable of generating data at data rates in excess of 5 Gbps and had flight heritage. The CMV4000 from CMOSIS was selected.

The main characteristics of the CMV4000 are listed below:

- Monochrome 2k x 2k pixels;
- Colour version available with 1k x 1k pixels;
- Global shutter;
- Exposure during readout;
- Correlated double sampling;
- Integrated ADCs;
- Selectable resolution of 10-bits or 12-bits;
- Multiple high dynamic range options;
- 16, 8, 4 or 2 channel LVDS outputs;
- On-chip programmable PLL for high-speed clock generation;
- Maximum frame rate of 180 frames per second;
- Successfully flown in space missions.

These characteristics make the CMV4000 ideal, in many ways, for the SpaceFibre camera. A maximum data rate well above the capability of a single-lane SpaceFibre interface is readily achieved, permitting experimentation with multi-lane interfaces to the camera.

V. SPACEFIBRE CAMERA

The SpaceFibre camera provides a high-resolution, high frame-rate camera which is suitable for both Earth Observation, vision-based navigation and robotic applications. It incorporates the Microsemi RTG4 FPGA which, as well as providing the image sensor interface, control logic and SpaceFibre interfaces, has plenty of room left for data compression or other image processing applications to be integrated in the camera. One specific example, of interest to the team at Dundee, is image feature extraction and tracking for vision-based navigation of planetary landers.

A block diagram of the SpaceFibre Camera is provided in Fig. 2.



Fig. 2. SpaceFibre Camera Block Diagram

There are ten connectors on the SpaceFibre camera: four SpaceFibre connectors (2-lane nominal and 2-lane redundant), two SMA trigger input connectors, two SpaceWire connectors (nominal and redundant) and +5V DC power connectors (nominal plus JTAG and redundant plus configuration).

The CMV4000 image sensor is configured and controlled by the RTG4 FPGA. The image sensor sends image data to the FPGA via 16 LVDS differential pairs running at up to 480 Mbps per pair. The FPGA includes four SpaceFibre lanes which can operate as two 2-lane links or one 4-lane link. When operating with two 2-lane links, one is active and the other is redundant. The FPGA transfers the image data out of the camera over the active SpaceFibre link. There are also two SpaceWire interfaces (nominal and redundant) which can be used for transferring data instead of using the SpaceFibre interfaces.

Camera configuration, control and housekeeping requests are received over virtual channel 0 of the active SpaceFibre link or over the active SpaceWire link when in SpaceWire interface mode. The FPGA interprets these commands and transfers information to and from the image sensor accordingly, using the control interface of the image sensor.

Attached to the image sensor is a bank of EDAC protected DDR memory which can be used to store images when the FPGA is being used as an image processor.

Power is provided to the SpaceFibre camera via a pair (nominal and redundant) of 15-pin micro-D connectors. The input power is DC-DC converted to the various power supplies required by the FPGA and image sensor. Power on reset circuitry is also provided to reset the FPGA and image sensor during power-up.

The JTAG interface for programming and debugging the FPGA is accessible using the spare pins of the nominal power connector. Six system configuration signals are similarly set using the spare pins of the redundant power connector. These six pins may also be used for debug input/output. The two SMA connectors provide two trigger inputs which are buffered using Schmitt triggers and passed to the FPGA.

A crystal oscillator is provided for the FPGA for general operation along with a programmable oscillator for the SpaceFibre links. A flexi connector is connected to spare pins of the FPGA for test and debug use.

The SpaceFibre camera electronics is implemented on a single flexi-rigid PCB as illustrated in Fig. 3. The blue areas represent the rigid part of the PCB and the green the flexible

part. The power supply circuitry is on one board, along with the SpaceWire connectors, and power supply/configuration connectors. The RTG4 is on the middle board, together with the DDR memory and the four SpaceFibre connectors, to minimise the PCB track lengths from the FPGA. The image sensor is on the third board, along with the power supplies for the image sensor. The LVDS lines from the image sensor run through the flexi part of the PCB to the FPGA.



Fig. 3. Arrangement of SpaceFibre Camera Functions on Flexi-Rigid PCB

A photograph of the flexi-rigid PCB or the SpaceFibre Camera is illustrated in Fig. 4.



Fig. 4. Flexi-Rigid PCB for SpaceFibre Camera

The pad grid for the FPGA can be clearly seen. The size of the FPGA 42.5x 42.5 mm was the main driver for the size of the camera. The PCB is folded with the FPGA facing down, the power board folded over the rear of the FPGA board and the image sensor board folded over the power supply board. This results in the image sensor being visible at the front of the camera and the FPGA being lid down at the rear. A thermal plastic layer conducts heat directly from the lid of the FPGA to the base on the rear of the camera.

The populated PCB can be seen in Fig. 5. There is a socket for the image sensor so that a monochrome or colour version can be fitted.

The housing for the camera was designed so that the PCB could be mounted on separate parts of the housing. The housing with the PCB is then folded up and held in position with some bolts. A CAD model of the camera was constructed, see Fig. 6., and this was then implemented as a 3D printed plastic housing to check the mechanical arrangement. The various mechanical parts were then machined in aluminium and the complete camera constructed. A C-mount was used to fit the lens to the camera housing. It is possible to shim the lens to provide the correct back focal length. The complete camera is shown in Fig. 8.



Fig. 5. Assembled SpaceFibre Camera PCB



Fig. 6. SpaceFibre Camera CAD Model



Fig. 7. SpaceFibre Camera Housing



Fig. 8. SpaceFibre Camera Partly Installed in Box

An initial test design was developed for the RTG4 FPGA which was used to check the correct operation of most of the elements of the SpaceFibre camera. A block diagram of this FPGA design is shown in Fig. 9. The image sensor interface is on the right-hand side of this diagram. A 100 MHz oscillator on the PCB provides the basic clock signal to the FPGA. This is fed through a global clock generator to provide the clocks required inside the FPGA. The bit and word clocks required by the image sensor are produced by a clock generator circuit and passed to the image sensor using LVDS signals. The image data from the image sensor is passed back on 16 LVDS signal pairs together with a control signal which indicates when the data on the data lines is valid and a clock which is synchronous with the data. The data is clocked into the FPGA by a camera readout circuit. The data is passed as sixteen12-bit data streams into a packet encapsulator. Each image frame is encapsulated in a separate packet. Header information is added to the start of packet and it is terminated by an end of packet market (EOP). The packet data is then passed to virtual channel VC1 of SpaceFibre interface SpFi1. It is passed through the SpaceFibre interface and Serialiser/Deserialiser (SerDes0) and out of the FPGA pins to one of the SpaceFibre connectors on the PCB. At present the image data is only connected to SpFi1, but in future it will also be connected to virtual channel VC1 of SpFi2, which will act as a redundant SpaceFibre interface or part of a dual-lane or quad-lane interface.

Virtual channel VC0 of the two SpaceFibre interfaces are connected to an internal SpaceWire router, which also connects to the two SpaceWire interfaces on the camera. The SpaceWire router connects to a configuration port (port 0 on the router) which is used to access the configuration, control and monitoring interfaces of the SpaceFibre interfaces, SpaceWire interface and the SpaceWire routing switch. Configuration is done using Remote Memory Access Protocol (RMAP) commands. Port 5 of the SpaceWire routing switch is connected to a second RMAP interface which is used to talk to the image sensor control interface. The image sensor is configured and controlled via an SPI interface. RMAP commands arriving over SpaceFibre or SpaceWire are converted into SPI commands which read and write registers inside the image sensor chip. The Frame Request control signals are also configured and controlled via the RMAP interface.

There is a DDR memory interface on the FPGA, but this has not yet been tested. It will be used for storing image data when performing image processing tasks in the RTG4 FPGA.

After initial testing and debugging the SpaceFibre camera was brought into operation. An image display application on a PC was able to capture and display the image data at a frame rate of around 19 Hz. For simplicity, each image pixel was packed into a sixteen-bit word. The resulting data rate is around 1.2 Gbps. The camera is capable of much higher frame rates and with a single SpaceFibre link it is expected that a frame rate of 20 Hz should be possible.



Fig. 9. SpaceFibre Camera Initial FPGA Design

VI. CONCLUSIONS AND FUTURE WORK

The SpaceFibre camera is operational and running well with a single SpaceFibre link. An image display application on a host PC is able to receive and display data at a frame rate of around 19 Hz, corresponding to a data rate of 1.2 Gbps. The data rate can be increased by simply changing some registers in the image sensor. The complete camera interfaces, SpaceFibre interfaces, SpaceWire interfaces, SpaceWire router and configuration logic take around 10% of the RTG4 FPGA leaving plenty of FPGA resources for the image processing applications.

Now that the image acquisition has been shown to be working, the next step in the project is to complete a more comprehensive FPGA design, which will permit redundant single-lane, redundant dual-lane and quad lane operation of the SpaceFibre interfaces. It will also incorporate the DDR memory and provide some example image processing applications on the RTG4.

As well as using the SpaceFibre interfaces for sending data from an image sensor to a mass-memory unit, it is possible to pass data between image sensors, since there are four SpaceFibre interfaces in each camera. The use of this capability for stereo-vision processing distributed between two cameras will be explored.

ACKNOWLEDGMENT

The research reported in this paper has been supported in part by the following organisations and contracts:

- The European Space Agency under ESA contract numbers 4000102641 (SpaceFibre Demonstrator) and 17938/03/NL/LvH (SpaceFibre), and Airbus DS subcontract number G010003963 (NGMMA).
- The European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement numbers 263148 (SpaceWire-RT) and 284389 (VHiSSI).

- The UK Space Agency and CEOI-ST under University of Leicester contract numbers RP10G0348A02 (SUNRISE), RP10G0348B206 (SpaceFibre-VPX) and RP10G0348A207 (SpaceDSP).
- The Centre for Earth Observation Instrumentation under University of Leicester contract numbers RP10G0327D13 (LOCUS Elegant Breadboard).

REFERENCES

- S. Parkes, A. Ferrer Florit and A. Gonzalez Villafranca, "SpaceFibre Standard", Draft K2, University of Dundee, December 2017.
- [2] S. Parkes, C. McClements and M. Suess, "SpaceFibre", International SpaceWire Conference, St Petersburg, Russia, 2010, ISBN 978-0-9557196-2-2, pp 41-45.
- [3] S. Parkes et al, "SpaceFibre: Multi-Gigabps Interconnect for Spacecraft On-board Data Handling", IEEE Aerospace Conference, Big Sky, Montana, 2015.
- [4] A. Ferrer Florit, A. Gonzalez Villafranca and S. Parkes, "SpaceFibre Multi-Lane", International SpaceWire Conference, Yokohama, Japan, 2016, ISBN 978-0-9954530-0-5.
- [5] ECSS Standard ECSS-E-ST-50-12C, "SpaceWire, Links, Nodes, Routers and Networks", Issue 1, European Cooperation for Space Data Standardization, July 2008, available from <u>http://www.ecss.nl</u>.
- [6] S. Parkes, "SpaceWire Users Guide", ISBN: 978-0-9573408-0-0, STAR-Dundee, 2012, available from <u>https://www.stardundee.com/knowledge-base/spacewire-users-guide</u> (last visited 11th Feb 2018).
- [7] S. Parkes et al, "SpaceWire and SpaceFibre on the Microsemi RTG4 FPGA", IEEE Aerospace Conference, Big Sky, Montana, 2016.
- [8] https://www.microsemi.com/.../135193-ds0131-rtg4-fpgadatasheet. Last visited 21st April 2018.
- [9] S. Parkes et al, "A Wideband Spectrometer in the Microsemi RTG4 FPGA", Data Systems in Aerospace (DASIA) Conference, Oxford, UK, May 2018.

SpaceVPX-RTG4 Board with SpaceWire or SpaceFibre Backplane

On-board Equipment and Software, Long Paper

Steve Parkes School of Science and Engineering University of Dundee Dundee, UK <u>s.m.parkes@dundee.ac.uk</u> Martin Dunstan, Chris McClements, Pete Scott, David Dillon STAR-Dundee Ltd. Dundee, UK

Albert Ferrer Florit, Alberto Gonzalez Villafranca STAR-Barcelona San Cugat, Spain

Abstract— SpaceVPX (VITA 78.0) is built on the ruggedized VPX standard. It addresses the need for redundancy in spaceflight systems and focusses on conduction cooled racks. SpaceVPX replaces the VMEbus control-plane of VPX with SpaceWire, while retaining the versatility of a user defined data plane serial interconnect. SpaceVPX-Lite (VITA 78.1) reduces the size and complexity of SpaceVPX. This paper describes a SpaceVPX-Lite board which uses SpaceWire and/or SpaceFibre for its backplane connections. The architecture of board is described along with its configuration options. An example application of the board as a wideband spectrometer is then described.

Index Terms—SpaceVPX, VITA, SpaceFibre, SpaceWire, Spacecraft Electronics, Radiation Tolerant, FPGA, RTG4, Spectrometer, FFT.

I. INTRODUCTION

There is a need for a standard backplane and rack for spacecraft avionics, to reduce development time and cost. However, each spaceflight system has its own requirements. Incorporating appropriate levels of redundancy is difficult and expensive in a standard system. Optimisations for mass, size and power savings also fight against a standard solution.

SpaceVPX-Lite is a draft standard (VITA 78.1) for spacecraft payload processing systems which defines a mechanical rack, an electrical interconnect including SpaceWire, a range of backplane profiles and a redundancy scheme. SpaceVPX and SpaceVPX-Lite have gone a long way to establishing an avionics standard or space applications.

SpaceWire is a widely used network technology for interconnecting payload data-handling equipment on-board a spacecraft. SpaceFibre is a new generation of SpaceWire spacecraft which has over ten times the performance and operates over both electrical and fibre optic media. It provides integrated quality of service and fault detection, isolation and recovery mechanisms which improve reliability and simplify redundancy. This paper describes a SpaceVPX-Lite board which uses SpaceWire and/or SpaceFibre for its backplane connections. The architecture of board is described along with its configuration options. An example application of the board as a wideband spectrometer is then described.

II. SPACEVPX-LITE

SpaceVPX (VITA 78.0) is built on the ruggedized VPX standard [1]. It addresses the need for redundancy in spaceflight systems and focusses on conduction cooled racks. SpaceVPX replaces the VMEbus control-plane of VPX with SpaceWire, while retaining the versatility of a user defined data plane serial interconnect. SpaceVPX-Lite (VITA 78.1) reduces the size and complexity of SpaceVPX [2]. It focuses on 3U sized boards, restricts and rationalizes the possible backplane configurations of SpaceVPX and concentrates on the support of utility, control and data planes.

A SpaceVPX-Lite system is illustrated in Fig. 1. . It has four types of module or board that plug into a backplane or rack:

1. System controller module, which controls the operation of the other modules in the rack. There are two system controller modules, one nominal and one redundant.

2. Payload module, which is the main processing or data-handling functions. There is a maximum of six payload modules in the rack.

3. Power supply module, which provides power to the power switch modules for distribution to the other modules in the rack. There are two power supply modules, nominal and redundant.

4. Power switch module, which distributes power from the nominal or redundant power supply to the system controller and payload modules.



Fig. 1. SpaceVPX-Lite Backplane

SpaceVPX-Lite separates the backplane communications into four principal planes or functions:

1. Control plane, which is used to pass configuration, control and monitoring information between the system controllers and the payload modules. SpaceWire is used for the control plane links [3,4]. SpaceFibre may also be used for the control plane connections [5,6].

2. Data plane, which is used to pass information between the payload modules. The data plane supports several possible multi-Gbits/s serial technologies, for example SpaceFibre or Serial Rapid IO.

3. Utility plane, which is used to pass power, clock signals and control/status signals between the various modules in the SpaceVPX-Lite rack.

4. Management plane, which is used by the system controller to control and monitor the power supplies and power switches.

III. SPACEVPX-RTG4

STAR-Dundee has developed a SpaceVPX-Lite board to help with its work on the VITA 78.1 standard [7]. The board was used to test and evaluate various concepts, several of which have now been incorporated into VITA 78.1.

The SpaceVPX-RTG4 board is designed to be able to operate as a SpaceVPX-Lite controller or payload module. When operating as a payload module the board can be programmed to perform many different payload processing functions. An FMC type slot is available on the board for adding analogue IO functions.

A block diagram of the SpaceVPX-RTG4 board is shown in Fig. 2. The main component on the SpaceVPX-RTG4 board is a Microsemi RTG4 FPGA [8,9]. The Microsemi RTG4 is a radiation tolerant FPGA with extensive logic, memory, DSP blocks, and IO capabilities. It is inherently radiation tolerant, having triple mode redundancy built in. The RTG4 has a flash configuration memory incorporated in the device. In addition, the FPGA incorporates 16 SpaceWire clock-data recovery circuits and 24 multi-Gbits/s SerDes lanes to support highspeed serial protocols like SpaceFibre.



Fig. 2. SpaceVPX-RTG4 Board

The RTG4 FPGA is connected to two banks of DDR memory with EDAC protection. There are two SpaceWire and two single-lane SpaceFibre connections on the front panel. An FMC connector is included on the board so that analogue IO functions can be added to the SpaceVPX-RTG4. Fig. 2. shows a dual ADC board which is capable of sampling I and Q signals at 2.4 Gsamples/s with 8-bit resolution.

The SpaceVPX-RTG4 board can be configured as either a system controller or a payload module. Its backplane connections include a control plane comprising either six SpaceWire links or six SpaceFibre links and a data plane of six SpaceFibre links.

When configured as a system controller, as illustrated in Fig. 2., the backplane connections comprise the following:

- Nominal and redundant system clock inputs which normally operated at 100 MHz;
- Radial clock outputs which distribute the system clock to each of the six payload boards using point-to-point LVDS connections;
- Nominal and redundant auxiliary clock inputs, which can be used for pulse per second or other timing signals;
- Radial auxiliary clock outputs which distribute the auxiliary clock to each of the six payload boards using point-to-point LVDS connections;
- Nominal and redundant system reset inputs;
- Radial system reset signals which distribute the system reset to each of the six payload boards;
- Up to four I2C buses which connect to the power supplies and power switch boards to control and monitor their operation;
- Six SpaceWire links, six single-lane SpaceFibre, or six dual-lane SpaceFibre links which connect to each payload module using point-to-point connections;
- Power input connections from the power switch boards.

When operating as a payload module the following connections are provided from the nominal and redundant system controllers:

- Nominal and redundant system clock inputs which normally operated at 100 MHz;
- Nominal and redundant auxiliary clock inputs, which can be used for pulse per second or other timing signals;
- Nominal and redundant system reset inputs;
- Nominal and redundant SpaceWire, single-lane SpaceFibre, or dual-lane SpaceFibre link.

In addition, there are power supply connections and SpaceFibre data plane connections between each of the payload boards.

A photograph of the SpaceVPX-RTG4 board is shown in Fig. 3. The board can be configured as a system controller or payload module using either SpaceWire or SpaceFibre as the backplane interconnect. This configurability is achieved using banks of resistors and capacitors, which can be seen to the left of the FPGA in Fig. 2. A dual ADC FMC board is fitted to the SpaceVPX-RTG4 board in Fig. 2.



Fig. 3. Photograph of SpaceVPX-RTG4 Board

IV. WIDEBAND SPECTROMETER

The SpaceVPX-RTG4 board has been used as the processing board in a microwave and Tera-Hertz radiometer being developed at Rutherford Appleton Laboratories [10][11].

The overall architecture of a spectrometer capable of analysing 8 GHz bandwidth is illustrated in Fig. 4. It is a hybrid spectrometer which comprises four individual wideband spectrometers (WBS V) which can each process 2 GHz bandwidth.

Each WBS V unit is a complete 2 GHz bandwidth spectrometer with I and Q analogue inputs, a 10 MHz (nominal) reference clock and a SpaceWire interface for configuration, control and monitoring and for retrieving the acquired power spectra. Several units can be interfaced to a SpaceWire router for connection via a single SpaceWire link to the spacecraft data-handling system. The SpaceWire router could be on the system controller board which controls all the WBS V payload modules and gathers science data from them. The resulting 8 GHz spectral data can be sent out of the nominal or redundant front panel SpaceWire or SpaceFibre interfaces of the system controller.



Fig. 4. Spectrometer Architecture

The architecture of an individual WBS V unit is illustrated in Fig. 2. and a photograph is shown in Fig. 5.



Fig. 5. WBS V Unit

The WBS V unit comprises two boards: the SpaceVPX-RTG4 board (shown in blue) and the FMC-3GHz-ADC board which is a dual ADC board (shown in green). The RTG4 FPGA contains the ADC interface, FFT, power spectrum, accumulation, control and monitoring and SpaceWire interface circuitry. The FMC board contains two high speed ADCs which are operated as an I and Q pair at 2.4 Gsamples/s.

The analogue I and Q signals from the RF front-end are connected to two SMA connectors on the FMC board. Each input to the ADCs is a single-ended (unbalanced) input through an edge mount SMA connector and a weak anti-aliasing filter. The anti-aliasing filter is for test purposes only, since an external anti-aliasing filter is expected for a flight unit.

The clock signals for the ADCs and FPGA are generated on board using a clock distributor and clock manager (CDCM) chip and a 1.2 GHz VCXO. The CDCM is locked to a 10 MHz reference clock from the FPGA. An external reference clock signal can also be used.

To operate the ADCs as an I and Q pair, it is necessary to ensure that the data outputs of the two ADCs are synchronised. This is done by resetting the data clocks of the two ADCs. This requires a Data CLK Reset signal which is synchronised to the 1.2 GHz clock requiring high speed logic. A photograph of the dual ADC FMC board is shown in Fig.



Fig. 6. Dual ADC FMC Board

Each ADC provides four de-interleaved data streams which are passed to the FPGA via the FMC connector. With a sample rate of 2.4 Gsamples/s the four data streams each operate at a data rate of 600 Msamples/s. These signals are transferred as low voltage differential signals (LVDS) to the FPGA. In total there are 128 data connections between the two ADCs and the FPGA. The four data streams are further de-interleaved within the FPGA.

A SpaceWire interface with support for the remote memory access protocol (RMAP) is provided for configuring, controlling and reading data from the WBS V. A trigger input or output interface is provided which may be used to trigger the operation of the WBS. A JTAG connector is provided on the SpaceVPX-RTG4 board for programming the Microsemi FPGA.

V. SPECTROMETER RESULTS

In this section the results of the WBS V are presented.

A. Development test set-up

A specially designed rack was used to house the WBS V during development and test, so that the two sides of the board were accessible to probe and monitor component operation. The operation of the WBS V spectrometer was tested using a signal generator connected to the inputs of the WBS V unit. The WBS V was controlled from a host computer via a SpaceWire link. The WBS V can be configured, controlled and monitored via SpaceWire and the results of data acquisition and processing can be retrieved and displayed.

B. Spectrometer results with signal generator input

Initially a complex 256-point FFT based spectrometer was implemented in the FPGA. The results of this spectrometer are illustrated in Fig. 7. showing a 1000 frame integration of a 541MHz I/Q signal using rectangular and Hann window functions. The largest peak is the input signal fundamental frequency and the three other large peaks are harmonics of the input signal fundamental. The noise floor is around 45 dB below the main signal.



Fig. 7. "First Light" power spectrum using 256-Point FFT of 541MHz tone

Following the success of the 256-point FFT, the 1024-point FFT was implemented. Tests were made with a signal generator at difference frequencies, windowing functions and integration times. The results of this design with two different window functions and 1 million frames of integration (0.4 seconds) are shown in Fig. 8. The largest peak is the input signal fundamental frequency and the three other large peaks are harmonics of the input signal fundamental. The noise floor is around 56 dB below the main signal.



Fig. 8. Power spectrum using 1024-point FFT with 1 million frame integration

C. Spectrometer results with a 360GHz receiver and molecular spectra

The WBS V unit sealed in its case was tested at Rutherford Appleton Laboratories (RAL) with a 360GHz receiver using nitrous oxide and methanol at different pressures to provide real signal testing. A photograph of the test configuration and block diagram are shown in Fig. 9.

The gun diode was adjusted to select the local oscillator frequency around 90GHz. The precise setting depended on the molecular sample whose spectra was being captured. Note that both upper and lower side bands of the nominally 360GHz signal are captured by the WBS V. The targets were selected so that there were no lines in the upper side band or were much weaker than those in the lower side band (LSB).



Fig. 9. WBS V with 360GHz receiver

The vacuum chamber into which the molecular samples were introduced is shown in Fig. 10. showing how it is observed by the 360GHz front-end and the location of the cold target (blue bucket). During the tests the marked movable flat mirror was rotated down to block the vacuum target from the front-end and to reflect the cold target instead. A corrugated black plastic insert was placed in front of the mirror to act as a hot (room-temperature) target.



Fig. 10. Vacuum chamber with cold targets, flat mirror and front-end

To capture spectra the chosen molecular sample was introduced to the vacuum chamber and the pressure adjusted to the required upper target. The WBS V was used to capture three spectra, typically using two second integration periods with the Hann windowing function. The first spectra are of the cold (liquid nitrogen) target in the blue bucket, the second of the hot target, and the third of scene (the vacuum chamber). The ratio of the difference in the power of the hot and cold targets (in each FFT bin) relative to the difference in their temperatures (nominally 291K and 77K respectively) allows the power in each bin of the spectrum of the vacuum target to be converted into a brightness temperature; the power of the cold target in each bin provides the temperature reference. Between each capture the flat mirror had to be moved and positioned by hand so there are likely to be differences in the hot/cold results each time the spectra were captured.

A plot of a nitrous oxide sample showing the effects of pressure on the line width are shown in Fig. 11. The spectra were captured using the WBS V.



Fig. 11. WBS V spectra of nitrous oxide line at different pressures

VI. CONCLUSIONS AND FUTURE WORK

The SpaceVPX and SpaceVPX-Lite standards have been introduced and the design of a SpaceVPX-Lite board which uses SpaceWire and/or SpaceFibre for its backplane connections has been described. SpaceFibre has been included in the SpaceVPX and SpaceVPX-Lite standards as control plane or data plane interconnect. The SpaceVPX-Lite standard is in the process of being completed.

An example application of the Space-VPX as a wideband spectrometer has been outlined. It is intended to extend the bandwidth of STAR-Dundee's spectrometer technology to 8 GHz.

ACKNOWLEDGMENTS

The research reported in this paper has been supported in part by the following organisations and contracts:

- The European Space Agency under ESA contract numbers 4000102641 (SpaceFibre Demonstrator) and 17938/03/NL/LvH (SpaceFibre), and Airbus DS subcontract number G010003963 (NGMMA).
- The European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement numbers 263148 (SpaceWire-RT) and 284389 (VHiSSI).
- The UK Space Agency and CEOI-ST under University of Leicester contract numbers RP10G0348A02 (SUNRISE), RP10G0348B206 (SpaceFibre-VPX) and RP10G0348A207 (SpaceDSP).
- The Centre for Earth Observation Instrumentation under University of Leicester contract numbers RP10G0327D13 (LOCUS Elegant Breadboard).

REFERENCES

- [1] VITA, "SpaceVPX System", ANSI/VITA 78.00-2015, April 2015.
- [2] Scott Goedeke, et al, "SpaceVPX Lite, Lightweight SpaceVPX Systems Specification", VITA 78.1, Draft revision 2.3, VITA, 14 July 2016.
- [3] ECSS Standard ECSS-E-ST-50-12C, "SpaceWire, Links, Nodes, Routers and Networks", Issue 1, European Cooperation for Space Data Standardization, July 2008, available from <u>http://www.ecss.nl</u>.
- S. Parkes, "SpaceWire Users Guide", ISBN: 978-0-9573408-0-0, STAR-Dundee, 2012, available from <u>https://www.stardundee.com/knowledge-base/spacewire-users-guide</u> (last visited 21st April 2018).
- [5] S. Parkes, A. Ferrer Florit and A. Gonzalez Villafranca, "SpaceFibre Standard", Draft K2, University of Dundee, December 2017.

- [6] S. Parkes et al, "SpaceFibre: Multi-Gigabit/s Interconnect for Spacecraft On-board Data Handling", IEEE Aerospace Conference, Big Sky, Montana, 2015.
- [7] S. Parkes et al, "A Prototype SpaceVPX Lite (VITA 78.1) System using SpaceFibre for Data and Control Plane", IEEE Aerospace Conference, Big Sky, Montana, 2017.
- [8] S. Parkes et al, "SpaceWire and SpaceFibre on the Microsemi RTG4 FPGA", IEEE Aerospace Conference, Big Sky, Montana, 2016.
- [9] Microsemi, "DS0131 Datasheet RTG4 FPGA", <u>https://www.microsemi.com/.../135193-ds0131-rtg4-fpgadatasheet</u>. Last visited 21st April 2018.
- [10] S. Parkes et al, "A Wideband Spectrometer in the Microsemi RTG4 FPGA", Data Systems in Aerospace (DASIA) Conference, Oxford, UK, May 2018.
- [11] S.P. Rea, et al, "The Low-Cost Upper-Atmosphere Sounder (LOCUS)", 26th International Symposium on Space Terahertz Technology, Cambridge, MA, 16-18 March 2015.

Networks & Protocols (Long)

Real Time Video Data Transmission in SpaceFibre Networks with the ESDP Transport Protocol

SpaceFibre, Long Paper

Alexey Khakhulin, Igor Orlovsky

Rocket and Space Corporation Energia after S.P. Korolev Korolev, Moscow region, Russia {Alexey.Hahulin, Igor.Orlovsky}@rsce.ru

Abstract— The possibility of different traffic (such as the commands, the real time video, the Best Effort traffic) simultaneous transmission is one of main requirements for modern aerospace networks. Generation of traffic can be periodical or aperiodical. The real time video traffic is one of important traffic types in SpaceFibre networks. This traffic is transmitted via network simultaneously with other traffic types (command traffic, Best Effort traffic). For real time video traffic limited jitter is an important constraint. In this paper we evaluate jitter of videoframes for different QoS in a SpaceFibre network.

We review the capabilities of the scheduling mechanism (SpFi datalink layer) for allocating of bandwidth between virtual networks with different data flow types and QoS. The focus is at streaming data communication. The usecase is motion video transmission in SpaceFibre networks with ESDP (Energia Streaming Data Protocol) transport protocol. We evaluate achievable delivery time and jitter as functions of the network size, timeslot duration, and the data units' size.

Index Terms — Spacecraft onboard networks, Streaming data, SpaceFibre, Motion imagery.

I. INTRODUCTION

Modern onboard networks are used for transmission of different traffic types. A significant part of traffic is real time video traffic. Guaranteed delivery time and jitter constraints are important requirements for this type of traffic. Jitter is difference between maximal packet delivery time and minimal packet delivery time.

One or several real time video flows and some flows of other types can be transmitted simultaneously in a SpaceFibre network. Transmission paths of different traffic can overlap. The competition of data flows in overlapping points causes jitter and, correspondingly, causes maximal packet delivery time increase.

Different formats of videoframes can be used. For some formats the length of all videoframes is fixed [1,2]. For other formats (e.g. JPEG) the length of videoframes can substantially (in several times) vary [3,4].

Yuriy Sheynin, Elena Suvorova, Ilya Korobkov, Valentin Olenev, Irina Lavrovskaya

Saint-Petersburg State University of Aerospace Instrumentation Saint Petersburg, Russia sheynin@aanet.ru, {ilya.korobkov, valentin.olenev}@guap.ru,

suvorova@aanet.ru, irina.lavrovskaya@guap.ru

Sequential videoframes generation interval between of sequential videoframes is typically constant, independently of videoframes format.

Further we consider transmission of real time video traffic with the ESDP transport protocol (formerly called the STP-2 protocol [5]) over a SpaceFibre network.

II. BRIEF DESCRIPTION OF THE ESDP

A. General description

The ESDP protocol is a transport layer protocol over SpaceFibre. It is designed for transmission of streaming data (for example video streams) with stable intensity between end nodes of a SpaceFibre network. It can be used for streaming traffic transition with PDUs of the same size (for example, data from sensors, video frames, or lines of uncompressed video) or PDUs of the variable size (compressed video).

Data packets are not buffered in the source, therefore hardware cost of implementation is low. Data packets are delivered without acknowledgements and retransmissions, because in case of loss or damage of a videoframe in the network transmission of newest videoframes is more relevant than retransmission of oldest videoframes. In addition retransmissions of oldest videoframes can lead to essential overloads in the SpaceFidre network and increase data delivery time.

The ESDP is a connection-oriented protocol. It ensures decrease of PDU transmission overheads due short transport headers of packets and control of packet parameters in the source (maximal allowable packet length and intensity of packets generation). This feature allows to protect the virtual network (and virtual networks with lowest priority) from faulty source.

The ESDP protocol supports up to 4096 transport connections for one device.

Implementation of ESDP may be either completely in hardware or hardware/software. To improve performance it is recommended to implement data send/receive ESDP mechanisms in hardware; the transport connections control could be done in software.

ESDP protocol has failure detection and indication mechanisms also. The protocol detects errors in header and in payload of packets; packet loss (for data packets and servise command/packets). It monitors also the failure of a master/slave device (as a result of the device crash or the communication link disconnection between them).

There are two main operating schemes:

1. Data exchange between two devices, one of which is the master and the second - slave. It supports data transfer from slave to master and from master to slave.

2. Data exchange directly between two slave devices (transmitter and receiver) under control of the third device – the master node.

III. TRANSMISSION OF REAL TIME VIDEO TRAFFIC

A. General issues

The SpaceFibre network can include one or several sources of real time video traffic. The parameters of videoframes (size and intensity), the requirements (delivery time and jitter) for different sources may vary.

Videoframes are transmitted via the SpaceFibre network in ESDP data packets. One videoframe can be placed in one or in several ESDP data packets (for example, each video line can be placed in a separate ESDP data packet). At the SpaceFibre Data Link layer every packet is divided into SpFi frames, as illustrated in fig.1. The boundaries of SpFi frames are not aligned to boundaries of packets.



Fig. 1. Transmission of a videoframe via SpaceFibre network

IV. TIME CHARACTERISTICS OF REAL-TIME VIDEOTRAFIC

A. General information

Let's consider causes of the packet transmission jitter and, correspondingly, growth of the maximal packet delivery time for real-time video traffic.

Every packet is transmitted over one Virtual network.

At the Data Link Layer the SpFi data frames that belong to different Virtual Networks compete with each other and with other types of information flows (Broadcast Frames, FCT, ACK|NACK) for data link (fig. 2 region 2). This competition occurs at the Data Link Layer in every output port (output port of the source node and corresponding output ports of transit routers).

Only packets, which belong to the same Virtual network can compete for resources on the Network layer. When several source nodes use the same Virtual network and the data paths from these nodes go through the same resources (the same output ports of routers) corresponding packets can compete for access to the output port (fig. 2 region 1). The spread of output port waiting time is a part of packet's delivery time jitter.



Fig. 2. The regions of competition of data flows in the router Consequently we can evaluate jitter by:

$$J_p = J_n + J_l \tag{1}$$

Where J_p – jitter for the packets of considered traffic J_n – part of jitter that appears at the Network Layer J_l – part of jitter that appears at the Data Link Layer The maximal delivery time of the packet can be evaluated

by:

$$T_{p\max} = T_{p\min} + J_p \tag{2}$$

Where T_{pmax} – the maximal packet delivery time T_{pmin} – the minimal packet delivery time

In this paper we consider in details jitter at the Data Link Layer

B. Jitter on the Data Link Layer

The priority of the Broadcast frames, FCT, ACK, NACK is higher than the priority of Data frames. Therefore percentage of Broadcast frames, FCT, ACK, NACK in the information flow determines jitter for data frames in all virtual channels.

It is not possible to eliminate completely this part of jitter due the transmission of Broadcast, FCT, ACK|NACK, which are asynchronous to the data transmission.

We denote jitter that occurs due to ACK|NACK in considered output port of router *i* in the transmission path of the traffic – T_{ai} . In this paper we do not consider multicast transmission of the real time video traffic, therefore in every router the considered traffic is transmitted via one output port.

Designer can control jitter due to Broadcast transmission by setting the percentage of bandwidth for Broadcasts. We denote this part of jitter - J_{bi} .

The data frames that belong to different virtual networks (virtual channels on the Data Link Layer) compete with each other for access to the data transmission medium. The order of data frames from different VC in transmission medium is determined by the VCs' settings (priority, reserved bandwidth, scheduling) and the used bandwidth. The jitter of SpFi data frames transmission time depends on these parameters. Let's denote this part of jitter for output port of

considered router J_{di}

Jitter occurs for every SpFi data frame individually on the Data Link Layer. We should consider jitter of every SpFi data frame for evaluation of the data packet jitter.

The jitter of a data packet in data link in the router *i* can be evaluated by:

$$J_{li} = (J_{ai} + J_{bi} + J_{di}) * N$$
(4)

Where J_{li} – the jitter of the packet in an output port of the router *i*;

N - the number of SpFi data frames for transmission of a ESDP data packet of maximal length.

We consider that the jitter of every SpFi data frame of one VC, is equal, because we evaluate the worst case.

C. The jitter at the Data link layer when bandwidth reservation is used

Let's consider J_{di} when bandwidth reservation is used (data from all VC can be transmitted in every timeslot at the Data Link Layer – the scheduling at the Data Link layer is not used).

We shall assume, that all data packets are transmitted with SpFi data frames of maximal size (256 data bytes when the Multilane does not used). It allows to simplify the formulas. We denote the transmission time of such SpFi data frame via SpFi line by T_f .

Let's consider causes of jitter when bandwidth reservation is used. The fig. 3 illustrates some possible transmission sequences of SpFi data frames from different Virtual channels.

The fig.3 a) shows the "ideal" variant – the Data link layer includes 4 VC, reserved bandwidth and priority for every VC is equal. The SpFi data frames from the Network Layer go to every VC with the same intensity. This intensity corresponds to the reserved bandwidth, the data frames to every VC goes in about the same intervals of time. As the result the frames from all VC are transmitted via data link and the interval of time between transmissions of sequential frames from one VC is identical. Interval of time between the placement of the SpFi data frame into VC buffer and transmission of this frame is very short.

In this case when the bandwidth percentage q_i is allocated to the Virtual network *i*, the packet transmission time over this virtual network via concrete output port can be evaluated by:

$$T_{pi} = . \frac{T_{p\min_i}}{q_i}, \qquad (3)$$

where T_{pmini} the time of packet transmission when the traffic in other VC's is not presented.

The fig. 3 b) shows not ideal variant. In this variant, likewise the previous variant, all four VC have the equal reserved bandwidth. Priority of VC1 is higher than priority of VC2,3,4. The VC3 has lowest precedence at the given moment of time. As the result the SpFi data frame form VC1 is transmitted first, then two frames from VC2 are transmitted (precedence of VC2 is lower than precedence of VC3 this time), then two frames from VC3 are transmitted (because other VCs have not frames for transmission), then one frame from VC1 is transmitted, then four frames from VC4 are transmitted (precedence of VC4 is lower than precedence of VC3 this time), after that the third frame from VC3 is transmitted. Is this case the interval of time between transmission of sequential frames from one VC vary significantly, therefore jitter increases. The actual bandwidth precedence in a long time period for every VC corresponds to its reserved bandwidth. But packet transmission time can be essentially bigger than T_{pi} .

If the data packet transmitted via link when no traffic in

other VCs is presented, all SpFi frames of the packet will be immediately transferred to each other, this situation is represented on fig. 3 b) for VC4. In such case the transmission



Fig. 3. The examples of possible interleaving of SpFi data frames from different VC in the link

At the time of arrival of the SpFi data frame, which belongs to the considered VC, a frame from another VC can be transmitted via data link (regardless of the parameters set to the virtual channels).

If at the time of arrival of the data frame of the considered VC, several data frames for transmission lie in VC with highest priority, the frames from VCs with highest priority will be transmitted before the frame from the considered VC.

Denote by N_{hi} the maximal possible quantity of frames in VCs with priority higher than priority of the considered VC for output port *i* in the path. Then the part of jitter that

appears due these frames is equal to $N_{hi}*T_{f}$.

If at the time of arrival of the SpFi data frame of the considered VC, in VC with the same priority but highest precedence several data frames for transmission lie, the frames from VCs with highest precedence will be transmitted before frame from the considered VC.

Denote by N_{ri} the maximal possible quantity of frames in VCs with precedence higher than priority of the considered VC for the output port *i* in the path. Then the part of jitter appear due to these frames is equal to $N_{ri} * T_f$.

Thus, Jd_{*i*} can be evaluated by:

$$J_{di} = T_f + N_{hi} * T_f + N_{ri} * T_f$$

= $(1 + N_{hi} + N_{ri}) * T_f$ (4)

Let's evaluate the part of the jitter at the Data Link Layer when the packet goes via several routers.

When the packet is transmitted via the first output port of data path (output port of the source node) jitter of every SpFi data frame of the VC is $Ja_0+Jb_0+Jd_0$. (delay of the frame is distributed between 0 and $Ja_0+Jb_0+Jd_0$).

Jitter in a following output port in the VC data path is $Ja_i+Jb_i+Jd_i$. Delay of SpFi data frame in the output port *i* does not depend on delay of the frame in previous ports.

Therefore we can evaluate jitter as:

$$J_{l} = \sum_{i=0}^{p+1} J_{li} = \sum_{i=0}^{p+1} (J_{ai} + J_{bi} + J_{di}) * N, \quad (5)$$

where p+1 – the number of output ports (output port of the source node and output ports of the routers) in the path of the traffic.

The fig 4 shows jitter for the packet with size equal to one SpFi data frame for different quantity of output ports in the path. The parameters of traffic, which compete with considered traffic in every output port are represented in the table.

	J(1)	J(2)	J(3)	J(4)	J(5)
Nh	0	1	1	1	1
Nr	0	0	1	2	3

These diagrams show that jitter grow dramatically when the quantity of compete traffic frames grows.





Jitter for data packets with length = 25Kbytes, fig. 5 grows dramatically with growing of competing SpFi data frames number, but for considered variants it is less than 12ms.



Fig. 5. Dependency between jitter and the number of transient output ports for different quantity of compete frames (packet length = 25Kbytes)

Jitter of packets with length = 2Mbytes is represented on the fig. 6. On these diagrams we may see the same trends (same ratios) as on the fig 5. However value of jitter is essentially higher, for worst case it is equal to 860 ms.

Considering a specific example of network configuration, where the parameters of data flows, with the known parameters of devises (such as buffers' sizes), more accurate jitter evaluation is possible. The jitter according to this estimation will be lower, than the jitter, that is evaluated with previous formulas due such factors as accumulation of SpFi data frames in the VC buffer when the data link is occupied by other flows. Such accumulation allows to avoid "slipping" of a data frame from a VC with lowest priority between data frames from the VC with high priority.



Fig. 6. Dependency between jitter and the number of output ports for different quantity of compete frames (packet length = 2Mbytes)

Also if it is known that the data packet flow over a high priority Virtual network has very low intensity and short packets (shorter than the maximal data frame length), its influence can affect not every SpFi data frame of the considered traffic. But the general trend remains.

D. Using scheduling to decrease jitter

Now consider scheduling mechanism when different timeslots are allocated to different types of traffic at the Data Link Layer and evaluate jitter in this case.

The scheduling works at the the Data Link Layer and at the Network Layer. The change of timeslots is done simultaneously for all ports in the same router. However, there is a question - whether change of timeslots on Data link layer should be simultaneous in all routers of a network or simultaneity is not required. Other question – should scheduling in the Network and scheduling at the Data Link layer be synchronous.

Scheduling at different layers of network can be coherent – duration of timeslots and moments of their change are the same at all layers; or non coherent - duration of timeslots and moments of their change can vary at different layers (and maybe in different routers) in the network.

Using of the coherent approach

First consider the coherent approach – all terminal nodes and routers in the network use the single system time. Timeslots change is synchronous in all routers (on the Data link layer) and in all terminal nodes (on the transport layer (packet generation corresponds to timeslots change) and on the Data link layer). Assume that timeslot duration is enough for transmission of data packet from source node to destination node. Only one source node may transmit data in the timeslot in a VN.

Such approach allows to almost completely eliminate competition between data flows from different sources. (Competition may occur only on timeslots borders and depends on synchronization accuracy.) In this case jitter at the Network layer is very small (practically equal to zero). Jitter on the

Data Link Layer also is small; it includes only T_a and T_b parts. The diagrams of dependency between jitter of a ESDP data packet and quantity of output ports in the transmission path for different packet length are represented in the fig. 7 (N – number of SpFi frames used for transmission of one packet via output port). The graph JC, N=10 corresponds to the packet length 2,5 Kbytes, etc, JC, N=4000 – 1 MBytes, JC, N=8000 – 2 MBytes





Jitter grows essentially with growing of packet length and growing of output ports number in the transmission path. But these graphs show also that jitter for all considered variants is rather small, less than 16 ms.

Now compare jitter for coherent scheduling and jitter for guaranteed bandwidth.

The fig. 8 shows the results for packet size 25Kbytes. On this figure the diagram marked JC corresponds to the variant with coherent scheduling, the graphs marked J(i) correspond to the variant with guaranteed throughput (for different number of data frames of compete traffic in one output port).

These graphs show, that jitter for coherent scheduling is ten times less than jitter for guaranteed throughput and with minimal number of compete traffic frames (one data frme) in every output port. For example, when the the number of output ports is 10, JC = 192, J(1) = 2304. This ratio grows dramatically with growing of compete frames number; when number of compete frames is 5, the ratio is more than 50.



Fig. 8. Dependency between jitter of SpFi packet and quantity of output ports in the transmission path when scheduling is used

These ratios are maintained for packets with other sizes also.

But this approach has several features, which essentially constrain possibility of its application:

1. The quantity of data sources in a SpFi network should be not more than 64.

2. The accuracy of clock synchronization in routers and terminal nodes in general case depends on size of SpFi network. There should be margins between timeslots, their size depends on accuracy of clock synchronization.

3. Duration of timeslots should be enough for transmission of a whole packet between the source and the destination node. But the length of packets and the length of transmission paths can be vary significantly for different types of traffic. The length of packets can vary essentially for one traffic (in JPEG, for example). It may lead to situation when a considerable part of time the network resources are idle.



Fig. 9. An illustration of non-efficient usage of network resources

4. The PDU generation period in different sources may vary essentially (for example the network includes video cameras with 24,25 and 66 videoframes per second); generation of scheduling tables may be very complicated or impossible.

Due to these problems there is a need to have an alternative to the coherent approach in networks.

Using of non-coherent approach

Scheduling at the Data Link Layer can be used to allocate the guaranteed percentage of bandwidth to virtual channels when non-coherent approach is applied. Several SpFi data frames that belong to one VC (or one Virtual network) are transmitted during the time slot. Let's call such group of frames a «segment». Long packets can be transmitted via Data Link layer by several segments. An example of such SpFi data frames transmission is represented in fig. 10. The time moments when SpFi data frames go from the Network layer to the corresponding VCs of the Data Link Layer are shown at the upper part of this figure. The sequence of frames in the data link is represented at the bottom part of the figure.



Fig. 10. The example of the SpFi data frames from different VC interleaving in data link when non coherent approach is used

In this case the guaranteed percentage of the channel bandwidth is reserved for every VC. Moreover in contrast with the basic mechanism of Bandwidth reservation in SpaceFibre in this case using of this percentage by other VC is impossible (when transmission of data form only one VC in every timeslot is allowed).

In comparison with coherent scheduling approach this approach allows to eliminate the relation between the duration of timeslots in the Data link layer and the packet transmission time. Thereby most restrictions of coherent scheduling approach can be eliminated.

Therefore, selection of timeslot duration and generation of a schedule table, decreasing of idle time for networks with different traffic types will be more easily done with the noncoherent approach.

Let's compare jitter for this approach and jitter for basic mechanism of bandwidth reservation on the Data link layer is used. In this consideration we assume that data from only one virtual channel can be transmitted during one time slot.

The main parts of guaranteed bandwidth mechanism jitter Nh_i *Tf μ Nr_i *Tf do not exist in case of non coherent scheduling. The part of jitter that occurs due to waiting when the current SpFi data frame is transmitted (Tf) can arise only on the time slots boundaries.

But there is a new part of jitter for non-coherent scheduling – waiting of corresponding timeslot. This part of jitter depends on time slots duration and the scheduling table (list of time

slots, in which corresponding virtual channel is allowed to send data).

To consider transmission of ESDP packet via one output port, first we assume that the whole videoframe in the source node is ready for transmission.

We assume that number of SpFi data frames in the packet is essentially bigger, than quantity of SpFi data frames that are transmitted during an epoch. Therefore, we can neglect the fact that quantity of SpFi frames transmitted during the first epoch (when the packet transmission is started) and the last epoch (when the packet transmission is finished) may be essentially less, that quantity of SpFi data frames transmitted in every middle epochs. First we consider the situation, when the timeslots allocated to the considered VC, are not consecutive in the scheduling table. In such case we can evaluate jitter by the formula:

$$Jd_{i} = (\frac{N}{L^{*}(Nt-1)} - \frac{N}{L^{*}Nt})^{*}$$
(6)
(Nt*64*Tf)

Where Nt - the number of SpFi data frames, which are transmitted during one time slot.

L – the quantity of time slots in epoch in which the considered VC can transmit data.

Second, we consider the situation, when the timeslots allocated to considered VC, are consecutive in the scheduling table (on the borders of consecutive timeslots allocated to one VC the SpFi data frame from other VC can not «slip»). In such case, we can evaluate jitter by:

$$Jd_{i} = \left(\frac{N}{L^{*}Nt - Nl} - \frac{N}{L^{*}Nt}\right)^{*}$$
(7)
(Nt*64*Tf)

Where Nl – the number of consecutive timeslot groups in the scheduling table that are allocated to the considered VC (if all allocated timeslots are consecutive, Nl=1).

The diagrams in fig. 11 show dependency between jitter and the number of transient output ports for different quantity of timeslots groups that are allocated to the considered VC. 16 timeslots in the epoch are allocated to considered traffic in this example. 20 SpFi data frames can be transmitted during one timeslot. The marked Js diagram, N=100 (25KBytes), Nt=20, L=16, Nl=1, corresponds to the variant when all 16 allocated timeslots are in one group; Nl=2 – all 16 tilmeslots allocated by 2 groups with 8 timeslots in each; Nl=4 – all 16 tilmeslots allocated by 4 groups with 4 timeslots in each and etc. The graph marked Js, N=100 (25KBytes), Nt=20, L=16 corresponds to the variant when all the 16 timeslots are placed in scheduling table separately.

These diagrams show that jitter is proportional to the number of groups.

The length of the packets in this sample is 25Kbytes; for other packet length the ratios are same.



Fig. 11. Dependency between jitter of packet and the number of output ports in the transmission path for different number of timeslot groups

Let's evaluate jitter for data transmission path includes several routers.

These routers form a pipeline where the SpFi data frames are transmitted. First of all the SpFi data frame should be placed into in the output port buffer of the corresponding VC in the source terminal node (1 stage of pipeline). After that, the frame should be placed into the Retry buffer of the output port (2 stage), then the frame should be transmitted over SpFi line to the next device (router or terminal node) and placed into the Retry buffer of the input port (3 stage); then the frame should be placed into the buffer of corresponding VC of the input port (4 stage). Then the frame should be transmitted via the Network layer into the corresponding VC buffer of the output port of the router (5 stage); and so on. The length of the pipeline depends on the number of routers in the data path. The rate of writing data into a Retry buffer and reading data from a Retry buffer should correspond to the data transmission speed in the SpFi lane. The rates between other stages of the pipeline could be higher than data transmission speed in the SpFi line.

Timeslots change scheme 1.

Consider that timeslots in all routers in the network change simultaneously.

When duration of timeslot is shorter than is needed for transmission of a SpFi data frame over the pipeline from source to destination, then transmission of the frame will stop at some point along the way until next timeslot that is allocated to this traffic. If duration of the timeslot is longer than time that is required for transmission of a SpFi data frame from source to destination node then still not all frames that have passed the first output port will reach the last output port of the data path in the same timeslot.

Thus, in both cases transmission of some part of the frames start in one timeslot, which is allocated to the considered traffics, and finish in next timeslot, allocated to considered traffis (until this timeslot these frames would be in VC buffers of transit routers) in both cases. Correspondingly, for evaluation of jitter in every output port we can use formulas (6) or (7). Since the SpFi data frames belong to one packet, which is transmitted from source to destination via pipeline, we can use the formula for evaluation part of jitter at the Data Link Layer:

$$Jl = \sum_{i=0}^{p+1} (Ja_i + Jb_i) * N + \sum_{i=0}^{p+1} Jd_i + Tw \quad (8)$$

Where Jdi is determined by the formula (6) or (7),

Tw – time of waiting for nearest timeslot allocated to the considered VC in the source node output port (the packet generation time in the source node is asynchronous to scheduling in the Data Link layer).

As we can see from the formulas (6), (7) jitter is inversely proportional to the number of timeslots that are allocated to the considered traffic. Doubling of timeslots allows to decrease jitter in two times. Jitter is inversely proportional to duration of timeslots, doublung of timeslot duration allow to decrease jitter in two times.

The diagrams of dependency between jitter and quantity of output ports for the packet length 25Kbytes are represented in fig. 12. (For other packet length, the ratios are similar). The marked by JC diagram corresponds to jitter for coherent scheduling; the marked by Js diagrams correspond to jitter for non-coherent scheduling (with different timeslot duration, different size of timeslots groups in the scheduling table). The marked by J(i) diagrams correspond to jitter for guaranteed throughput with different number of compete frames.

These diagrams show that Js, L=16, Ln=1 (16 timeslots allocated to considered traffic consequently in scheduling table) is practically equal to JC. Diagrams show also that jitter for non-coherent approach (L=8 and Nt=20) is not substantially greater than jitter for the coherent approach. Jitter for non-coherent approach is less than jitter for guaranteed throughput J(2). Jitter for non-coherent approach does not depend on quantity of other traffic in the SpFi network unlike jitter for guaranteed throughput, that strongly depends form this parameter.

When duration of the timeslot is relatively short, the implementation of this scheme require very accurate synchronization. This requirement can significantly limit usage of this scheme in network with big number of routers.



Fig. 12. Dependency between jitter of packet and quantity of transient output ports for coherent scheduling, non-coherent scheduling and guaranteed bandwidth

Timeslots change scheme 2.

Let us consider situation when timeslots in different routers change asynchronously.

We can evaluate the minimal number (worst case) of SpFi data frames transmitted via the source node output port during first timeslot that is allocated to the considered traffic by the formula:

$$Nm_0 = \min(Nt, Nb1) \tag{9}$$

where Nb1 is the number of frames that can be placed in the corresponding VC buffers of input and output ports of the first router in the path.

Indded, if during transmission of the first segment of data in both source node and first router timeslot is allocated to the considered VC, then Nt data frames will be transmitted. If in the router runs another timeslot that belongs to other traffic, then only Nb1 frames will be transmitted, and after that all buffers of corresponding VC in the router will be occupied.

Assume, that buffer size of virtual channels, which belongs to the considered Virtual network, is same for all transient routers.

In each subsequent output port:

$$Nm_i = \min(Nt, Nb_{i-1}, Nb_i)$$
 (10)

 $Nm_i = \min(Nt, Nb_{i-1}, Nb_i)$ (10)

Then the jitter in one output port can be evaluated by:

$$Jd_{i} = \left(\frac{N}{L^{*}(Nm_{i}-1)} - \frac{N}{L^{*}Nt}\right)^{*}$$
(11)
(Nt * L * Tf + Nt * (64 - L) * Tf)

The packet jitter at the Data Link Layer can be evaluated by the formula (8) in which the value Jdi by formula (11) is used.

Jitter for this scheme has an intermediate position between jitter for scheme (1) and jitter when guaranteed throughput is used. Jitter for this scheme strongly depends on ratio between timeslot duration (number of frames transmitted in one timeslot via the output port) and buffer size in VC.

V. CONCLUSION

In the paper we presented evaluation of jitter for real time videoflows transmission when different QoS mechanisms are used at the Data Link Layer of the SpFi standard.

We show that when a Virtual network includes several data sources the network part of jitter can be in some times bigger than minimal packet transmission time if the scheduling in the Virtual network is not used.

We show that jitter at the Data Link Layer can be several times bigger than minimal packet transmission time if the scheduling at this layer is not used.

We show that scheduling in a Virtual network and at the Data Link Layer in the coherent mode allows to reach jitter practically equal to zero, but this approach has several essential implementation constraints and could be not applicable for many systems. We give the scheme for selection of the timeslot duration that allows to reach the appropriate jitter when non-coherent mode is used.

ACKNOWLEDGEMENT

The research leading to these results has received funding from the Ministry of Education and Science of the Russian Federation within the project part of the federal assignment under contract N 8.4048.2017/4.6 from 31.05.2017.

REFERENCES

1 ARINC Inc., ARINC Specification 818-2 (ARINC 818 Supplement 2), 2013. 131 p.

2 CCSDS Digital Motion Imagery Standard 766.1-B-1, 2015. 41 p.

3 JPEG - Next-Generation Image Compression (JPEG XL) Revised Draft Call for Proposals". Jpeg.org. Retrieved 1 March2018.

4 Overview of JPEG". jpeg.org. Retrieved 2017-10-16

5 Streaming Services over SpaceFibre Networks / Ilya Korobkov, Elena Suvorova, Yuriy Sheynin, Valentin Olenev // Proceedings of the 7th International SpaceWire Conference 2016, 151 – 158 pp.

SpaceWire and SpaceFibre Overview and Roadmap

PAPER NOT PUBLISHED

A SWAPC Scalable High Performance Hardware Standard for 6U/3U VPX Based Processing and I/O Systems

PAPER NOT PUBLISHED

Evaluation and Interoperability Testing of the SpaceWire-R Protocol

Networks and Protocols, Long Paper

Wojciech Mich, Krzysztof Romanowski, Piotr Tyczka ITTI Sp. z o.o. Poznań, Poland {Wojciech.Mich, Krzysztof.Romanowski, Piotr.Tyczka}@itti.com.pl Rafał Renk Adam Mickiewicz University Poznań, Poland Rafal.Renk@amu.edu.pl

Vangelis D. Kollias, Nikos Pogkas TELETEL SA Athens, Greece {V.Kollias, N.Pogkas}@teletel.eu

Abstract-SpaceWire-R is a transport-layer protocol that provides reliable data transfer capabilities to SpaceWire networks. The ESA-sponsored project SpaceR delivered the first European implementation of the protocol. The paper gives results of a number of tests devised to establish the conformance of the implementation to the protocol specification, to evaluate the protocol's performance with respect to different data characteristics and protocol parameters, and to show usability of the protocol in example scenarios. Further, tests showing interoperability of two independent SpaceWire-R implementations, the European one and one from JAXA and NEC, are presented. Finally, extensions to the protocol and tuning of selected protocol parameters are proposed.

Index Terms— SpaceWire, SpaceWire-R, Network Protocols, Reliable Data Transfer, Interoperability

I. INTRODUCTION

Reliable end-to-end data transfer is vital for many applications, and is provided mostly by transport-layer protocols, notably the TCP [1] of the TCP/IP family in ground applications. The SpaceWire protocol, increasingly used for payload data in space missions, lacks higher-layer advanced features needed to support transfer reliability. To fill that gap the SpaceWire-R protocol has been proposed [2,3].

Following the studies and evaluations performed by the protocol originators [4-6], a European project 'SpaceR', sponsored by ESA, was undertaken [7-8] for independent implementation and evaluation, with a view to standardization of the protocol by ECSS. In an extension of the project, interoperability tests of the SpaceR implementation and the implementation developed by JAXA and NEC were run.

The SpaceR project was jointly carried out by ITTI and TELETEL. ITTI was responsible for the development of the software implementation of the SpW-R (SpaceWire-R) protocol, validation, and testing. TELETEL provided its iSAFT

PVS Recorder/Simulator [9] and developed a new API (Application Programming Interface) to the iSAFT Simulator, based on TCP sockets.

This paper presents the main results obtained in SpaceR in terms of functional testing, performance evaluation, and interoperability testing of the SpW-R protocol based on the developed implementation. The paper also proposes and discusses some extensions and tuning of the protocol.

The paper is structured as follows. Section II gives a brief overview of the SpW-R protocol. The implementation of the protocol elaborated in the project and the testing platform developed for SpW-R testing and validation are described in Section III. Conformance and performance results are presented in Section IV. Section V summarizes the interoperability testing approach and results. In Section VI possible extensions and tuning mechanisms for SpW-R are proposed. Finally, Section VII contains concluding remarks.

II. OVERVIEW OF THE SPACEWIRE-R PROTOCOL

The SpaceWire-R communications protocol [3] is intended to provide on-board applications with reliable data transfer services over SpaceWire networks. The main functions of SpW-R are retransmission control (reliable transfer), multiplexing, segmentation, flow control, and keep-alive (heartbeat).

The protocol operates on top of the SpaceWire protocol and below the application layer. A SpW-R packet is sent as the cargo of a SpaceWire packet – preceded by the SpaceWire destination address and followed by the end-of-packet mark. The protocol ID field differentiates SpW-R packets from other SpaceWire traffic.

Key notions used to describe the protocol are the Transport End Point (TEP) and the transport channel. The TEP is defined as a point in a SpaceWire node that transmits (Tx TEP) or receives (Rx TEP) application data using a transport channel over a SpaceWire network. The transport channel is a protocoldefined one-way logical data path between a Tx TEP and an Rx TEP. There is only one transport channel between a Tx TEP and an Rx TEP; in other words, each TEP is dedicated to a certain channel. There can be, however, multiple transport channels between any pair of nodes. This implies that multiple Tx and Rx TEPs can co-exist in a node.

The procedures performed by the protocol entities can be classified into three categories:

- procedures at a Tx TEP, including channel control, segmentation, and transmission control
- procedures at an Rx TEP, including channel control, reconstruction, and reception control,
- common procedures at a node, including channel multiplexing and de-multiplexing.

The TEP procedures communicate with an application sending or receiving the data in the form of Service Data Units (SDU). An SDU can be segmented at the Tx TEP, with each segment sent in a separate SpW-R packet, and reassembled at the Rx TEP. Transmission control is based on positive acknowledgements (ACK) with retransmissions and a sliding window of packet sequence numbers.

III. IMPLEMENTATION AND TESTING PLATFORM

The system developed in SpaceR consists of two major components:

- an implementation of the SpW-R protocol,
- a testing platform for verifying and validating the implementation.

Figure 1 shows the main elements of the SpaceR system together with communication paths between the elements and the user (operator). The core applications of the testing platform include *Tester*, *Transmitter*, *Proxy Server*, *Link Emulator*, *Analyzer*, *SDU Generator*, as well as test scripts. The implementation of SpW-R is incorporated in the Transmitter application. The detailed scheme of the system is depicted in Fig. 2.



Fig. 1. Architecture of the system developed in SpaceR



Fig. 2. Detailed scheme of the SpaceR testing system

The SpaceWire network is handled by the Proxy Server, which provides its iSAFT Client API connection to other testing platform applications requiring a connection to a SpaceWire network (i.e. the Transmitter and the Link Emulator). The maximum number of proxy clients connected to the Proxy Server is equal to the number of the SpaceWire ports in the iSAFT PVS.

The Tester application runs, controls, and communicates with all other applications of the testing platform. Like the other applications of the testing platform, it uses a Python console to communicate with the user and to read and execute Python modules (Python script files). Each test scenario is represented by a script file, which is executed by this application. The result of each script is displayed on a screen in a form of a pass or fail message. In the latter case, a reason of the failure is also displayed. The detailed results of all scripts executed by the Tester are saved to a log file.

The Transmitter is an application that incorporates the protocol implementation as an internal node. The results of protocol operations are logged by an extensive logging subsystem. They are then used by the Analyzer utility, which extracts values of interest and generates an output file with statistics on the transmit channel and information on each transmitted SDU.

The implementation of the SpW-R protocol at a SpaceWire node stems from the core procedures described in the protocol specification [3]. Accordingly, three main implementation modules were developed that correspond to the protocol procedures mentioned in Section II and a utility module used by all of them.

The SpW-R implementation provides function calls and signals in accordance with the primitives defined in the protocol specification. These are:

- functions:
 - ChannelControl.request(ChannelNumber, DirectiveType),
 - DataTransfer.request(ServiceDataUnit,SduId, ChannelNumber),
- signals:
 - ChannelControl.indication(ChannelNumber, NotificationType),
 - DataTransferNotify.indication(SduId, ChannelNumber, NotificationType, Reason),
 - DataTransfer.indication(ServiceDataUnit, TransportChannel).

The SpW-R implementation is written in C++ and hosted on a Linux operating system.

IV. CONFORMANCE AND PERFORMANCE TESTING

Conformance and performance testing of the SpaceR platform and SpW-R implementation use several network setups, depending on whether a single SpW-R channel or multiple channels are employed in a test and whether errors are to be introduced by using the Link Emulator (see Figs. 3-5). The basic test setup uses the TELETEL iSAFT PVS as the interface to the SpaceWire network. For multiple channel tests the 4Links SRS Router is used. In addition, for some of the test scenarios an alternative interface based on the STAR-Dundee SpaceWire Brick [10] is used for extra tests.



Fig. 3. Basic SpaceWire network setup for conformance and performance tests

A. Conformance tests

The conformance tests include the following groups:

- basic operations:
 - channel opening,
 - data transfer,
 - channel closing;
- effects in open channels:
 - regular transmission,
 - heartbeat traffic,
 - abnormal conditions (e.g. disconnecting during data transfer);

• packet structure inspection.



Fig. 4. SpaceWire network setup for tests with the Link Emulator



Fig. 5. SpaceWire network setup for tests with multiple channels

Both single and multiple channel operation is subject to testing. Multiple channel tests involve two simultaneously operating SpW-R channels between node pairs, e.g. $A \rightarrow B$ and $B \rightarrow A$, $A \rightarrow B$ and $A \rightarrow C$, $A \rightarrow C$ and $B \rightarrow C$ (cf. Fig. 5), or two channels on the same pair, $A \rightarrow B$.

The results were established based on the extensive logs, processed by the Analyzer application. The packet structure was examined with the Wireshark application built in the iSAFT PVS platform. All the conformance tests were successful [8]; the protocol operated consistently with the specification.

B. Performance tests

These tests measure basic performance characteristics, including:

- throughput,
- packet latency, round-trip time, and jitter,
- number of retransmissions,

with respect to a number of varying data characteristics and protocol parameters, such as:

• SDU size,

- maximum length of application data field,
- sliding window size,
- single or dual channel activity,
- presence or absence of heartbeat,
- SpaceWire line rate,
- packet loss probability
- maximum retry count,
- transmit timer initial value.

A selection of results obtained in the performance tests is presented in Figs. 6-11.



Fig. 6. SDU throughput for different SDU sizes

Figure 6 shows the baseline result of the SDU throughput, obtained by sending series of 1000 SDUs with random contents in a setup of Fig. 3, with the SpaceWire line rate of 100 Mb/s, sliding window size = 8, and no segmentation. For the calculation of the SDU throughput, only the SDU bytes are counted, without the extra bytes of the other SpW-R packet fields or the SpaceWire address path. The throughput approaches the maximum possible value of 80% of the line rate for very large packet sizes.

Figure 7 shows that the software implementation and the hardware elements a packet has to go through before arriving at the receiving node (cf. Fig. 2) cause a latency of about 2.5 ms for SDUs of small and medium sizes. Large SDUs experience more latency due to the time required to move and process them.

The following figures present the dependency of the SDU throughput on the SDU size with varied other parameters or conditions. The length of the SDU series is 1000 except for the segmentation test (Fig. 10); the sliding window size is 128.

With different line rates (Fig. 8), throughput relative to line rate is higher for smaller rates, since the software processing overhead does not depend on the line rate.

Figure 9 shows the performance degradation coming from operating two channels simultaneously on the same network link and the same network nodes. The dual channel case involves concurrent channels $A \rightarrow B$ and $B \rightarrow A$ (cf. Fig. 3).



Fig. 7. Packet latency for different SDU sizes



Fig. 8. SDU throughput for different SDU sizes and SpaceWire line rates



Fig. 9. SDU throughput for different SDU sizes and channel numbers (P2.1: single channel; P2.2: two channels, results shown side by side)

The effect of SDU segmentation on throughput due to segmentation and reassembly overhead is presented in Fig. 10.

Figure 11 presents throughput for different sliding window sizes, ranging from 1 to 128 (the maximum defined in the protocol specification).







Fig. 11. SDU throughput for different SDU sizes and sliding window sizes

C. Usability demonstration

A demonstration of the operation of the protocol was performed at ESA/ESTEC, with two SpW-R nodes connected as shown in Fig. 12, the connection including a triplicated (redundant) segment. Logical addressing was used, with Group Adaptive Routing configured on the switches with redundant connections. Both SpW-R nodes were simultaneously sending and receiving a continuous stream of packets in two SpW-R channels, while one or more of the redundant links were being disconnected and reconnected and, consequently, some SpaceWire packets were truncated. At the application level, not a single data unit was lost in transmission, as long as at least one link of the group was active. Likewise, there was no application-level data loss on complete disconnection of the transmission path, provided the connection was restored before all retries up to the configured maximum number were exercised by the SpW-R nodes.



Fig. 12. SpaceWire network setup for the Group Adaptive Routing demonstration

As an auxiliary application for network discovery and configuration, the SPACEMAN Network Management Tool [11] ran on the iSAFT PVS connected to the SpaceWire network. The network was discovered by SPACEMAN and its representation displayed on the SPACEMAN console (Fig. 13). Different protocols: SpW-R, NDCP (Network Discovery and Configuration Protocol), and RMAP (Remote Memory Access Protocol), the latter two used by SPACEMAN for network discovery, operated correctly on the same network simultaneously.



Fig. 13. Group Adaptive Routing demonstration network as discovered by SPACEMAN

V. INTEROPERABILITY

Interoperability testing of independent implementations is an important requirement for a protocol to be standardized. SpaceR included tests of the interoperability of the protocol implementation developed in the project with an earlier implementation from JAXA and NEC [12].

Like SpaceR, the JAXA/NEC implementation is also software based. The system platform is different: the SpW-R stack is written in C and hosted on a SpaceCube 2 [13] with the T-Kernel operating system. Also different is the connection of the TEPs to the SpaceWire network: while the SpaceR implementation uses a single common connection for the Tx TEP and the Rx TEP, JAXA/NEC has dedicated connections for the Tx TEP and the Rx TEP to two separate SpaceWire ports of the SpaceWire switching router that is embedded in the SpaceCube. Figures 14 and 15 present the network setups used for testing the interoperability, and Fig. 16 shows the interoperability testbench.



Fig. 14. SpaceWire network setup #1 for interoperability tests



Fig. 15. SpaceWire network setup #2 for interoperability tests

Interoperability testing included transmissions between two SpW-R nodes (ITTI and JAXA/NEC) in both directions, with basically the same approach as with testing the conformance of the SpaceR implementation: channel control operations, data transfer, abnormal conditions, and examination of the structure of captured packets of various types, with the basic traffic and protocol parameters swept across a range of values.

The protocol operation was correct and the two implementations found to be interoperable, with few parameter adjustments needed, related primarily to packet addressing. One interoperability issue was found, regarding the heartbeat functionality. The SpaceR implementation follows version 0.4 of the protocol specification [3], while JAXA/NEC is compliant with version 0.3 [14]. These two specification versions are not compatible with regard to heartbeat packet sequence numbering convention. As a result, the ITTI node was discarding JAXA/NEC heartbeat packets as erroneous; a simple fix allowed accepting the packets and interoperability was obtained.



Fig. 16. Interoperability testbench at JAXA

VI. PROPOSED EXTENSIONS AND TUNING

Based on the experience from the implementation and the tests, some extensions to the protocol can be proposed. They are transparent to nodes that do not implement them: the extensions either use the secondary header or are limited to parameter optimization at a single node, which in general does not require direct interaction of the other node of the SpW-R channel.

A. Secondary header format

The protocol specification leaves the format of the secondary header as project-specific. However, some categories of information can be common to multiple projects and sharing it between the nodes could be useful. In particular, values of most of the parameters that define characteristics of a SpW-R channel do not have their place in the protocol data units and are specified by means of off-line tables, for which there is no 'in-band' communication foreseen. This may lead to practical problems of a node not knowing the parameters of its counterpart. One way of conveying that information is by encapsulating it in a SpW-R packet, at least at the channel opening time, and possibly also during a connection lifetime. The secondary header may be a place to carry that information.

Since different projects can need the secondary header for their specific purposes, a standardized method of identifying the secondary header use and format is needed. This can be patterned after e.g. JAS RDDP [15], where the initial byte of the secondary header is used for that purpose. Therefore it is
proposed to introduce the secondary header format ID and to assign an ID for a general-use header format, leaving it for specific projects to introduce more formats as needed.

The proposed general-parameter secondary header format, called the 'General Option Format', includes a list of SpW-R-related parameters, structured in a standardized way. This structure can e.g. follow the TCP options format, as a list of items where each item is composed of a one-byte ID, a one-byte option length, and the bytes of the option value, with the list terminated by a special-type option.

The proposed format of the secondary header is presented in Table I, and the format of a secondary header option - in Table II.

TABLE I. Secondary header by tes for the 'General Option Format'

Byte number	Name	Value
0; most significant half	Secondary Header Type	2
0; least significant half	Secondary Header Format Version ^a	0
1	List of Secondary Header Options, stored	contiguously

a. Cf. the JAS Secondary Headers [15]

 TABLE II.
 Secondary Header Option bytes for the 'General Option Format'

Byte number	Name	Value
0	Secondary Header Option Type	1-byte unsigned integer
1	Secondary Header Option Length	1-byte unsigned integer The length is measured in bytes and includes the Secondary Header option Type byte, the Secondary Header Option Length byte, and the Secondary Option value bytes
2 and following	Secondary Option Value	Specific to the particular option type

The collection of secondary header option types should reflect common needs of different projects. Some proposed types are listed in Table III. In addition to SpW-R protocol parameters, they include options that are not part of the protocol specification, which may be useful for protocol tuning or debugging. A complete list, including most of the protocol parameters, has been compiled in the SpaceR project [8].

TABLE III. SECONDARY HEADER OPTION TYPES

Туре	Length	Name	Value
0	2	End-of- Option-List	none
1	3	Priority Level	1-byte unsigned integer
2	10	Transmit Timestamp	Time of passing the packet containing this secondary header to the transmitting procedure below SpW-R. The value is an 8-byte integer holding time in nanoseconds since a reference start time

Туре	Length	Name	Value
3	10	Acknowledge d Timestamp	(ACK packets only) The transmit timestamp of the packet that is being acknowledged, copied from the secondary header option of the acknowledged packet. The value and reference time are like for the transmit timestamp
4	6 or 10	SDU ID	4- or 8-byte unsigned integer
5	6 or 10	Packet ID	4- or 8-byte unsigned integer
6	6	Maximum Length of SDU	4-byte unsigned integer
7	6	Maximum Length of Application Data Field	4-byte unsigned integer

B. In-band channel parameter control

If the secondary header format is implemented as proposed, channel parameters are communicated between both TEPs of the channel. Such communication can be performed not only at the channel opening time, but also during channel operation, thus making it possible to dynamically update the parameters and communicate the changes between the TEPs. Such changes could be made e.g. to channel priority, timers, or window and segment sizes. The secondary headers used for this purpose do not induce more traffic than necessary, as they are intended to be included in packets only when there is a change to communicate (unless timestamps or SDU/packet IDs are used), and to contain only the options that are relevant for the change.

The parameter values communicated in secondary headers can clearly be of different values than the current ones. It is thus necessary to have a possibility of controlling the action to be taken when a secondary header is read with channel parameters inconsistent with the current values. It is proposed that this dynamic parameter control is a feature that can be explicitly allowed or forbidden by each of the TEPs at the channel opening time, by including a new 'In-band Channel Parameter Control' option for the channel-opening control request. This option can be set to:

- 'no': ignore the new values;
- 'yes': change the parameters according to the new values.

The SpW-R implementation developed in SpaceR provides in-band channel parameter control. Most of the protocol parameters are sent by Tx TEP as secondary header options. The Rx TEP either uses them to update its set of parameters or ignores them, depending on the value of the In-band Channel Parameter Control option.

C. Dynamic protocol parameter tuning

1) Adjustment of the transmit timer based on round-trip time estimation

The transmit timer should be tuned for the particular network's round-trip time characteristics, so as not to cause unnecessarily long delays before retransmitting unacknowledged packets and, on the other hand, not to initiate unnecessary retransmissions. The tuning can be done in advance and set as a static value for a channel. However, with channels that use SpaceWire links in common with other SpW-R channels (or with other, non-SpW-R traffic), the round-trip time may experience changes that justify more refined tuning than adopting the worst case value.

A tuning mechanism similar to that used in TCP networks [16] is proposed as an optional functionality of a SpW-R TEP. Several algorithms are possible for the round-trip time estimator and the transmit timer adjuster. The SpaceR SpW-R implementation provides this mechanism as an option; the round-trip time estimator and the transmit timer adjuster use an algorithm based on the 'classical' TCP method [1].

2) Adjustment of the maximum retry count and the maximum length of application data field based on error rate estimation

This mechanism can react to changes in the packet error rate by varying the maximum retry count accordingly, so as to have more transmission attempts when the error rate increases; or by varying the maximum length of application data field, so as to send short packets when the error rate is high.

The SpaceR SpW-R implementation includes a preliminary test version of the mechanism. The error rate is estimated based on the number of packet retransmissions counted for a sliding window of packets (which is different from the SpW-R sliding window). The adjuster modifies the maximum retry count according to the trend in the estimated error rate.

VII. CONCLUSIONS

The paper presented the main outcomes of the SpaceR activity that aimed at delivering the first European implementation of SpW-R and validating the protocol. In particular, the results of the conformance and performance tests were shown. The SpaceR implementation of SpW-R is compliant with the protocol specification, and the specification itself is consistent and specifies completely and correctly the operation of the protocol. The results of the performance tests provide the behavior and quantitative evaluation of basic performance characteristics with respect to different data characteristics and protocol parameters. The tests show that for large packets it is feasible to achieve an SDU throughput close to the maximum SpaceWire throughput possible for a given line rate.

Interoperability testing of the SpW-R implementations coming from SpaceR and from JAXA/NEC was successful. The two implementations, which are completely independent, the protocol specification being the only common information involved in both, fully interoperated in the various setups examined. The successful interoperability tests and demonstration make a significant step towards prospective standardization of SpW-R by ECSS.

Finally, a few extensions to the protocol and tuning of selected protocol parameters were proposed. These propositions require further examination in order to assess their usefulness and the performance improvements they can bring.

ACKNOWLEDGMENTS

This work has been funded by the European Space Agency under the Polish Industry Incentive Scheme, contract no. 4000112693/14/NL/CBi. The authors would like to express their thanks to Mr David Jameux (ESA) for advice and support as the project's technical officer, as well as to Professor Masaharu Nomachi (Osaka University), Dr Keiichi Matsuzaki (JAXA), Dr Hiroki Hihara (NEC), Professor Seisuke Fukuda (JAXA), Mr Takayuki Ishida (JAXA), Mr Keita Tanaka (NEC), and Mr Takayuki Tozawa (NEC) for providing the JAXA/NEC implementation of the SpaceWire-R protocol and for participation in the interoperability tests.

REFERENCES

- J. Postel (ed.), Transmission Control Protocol. DARPA Internet Program Protocol Specification. RFC 793, USC/Information Sciences Institute, September 1981.
- [2] T. Yamada, "Proposal for SpaceWire-R," 17th SpaceWire Working Group Meeting, Noordwijk, December 2011.
- [3] T. Yamada, SpaceWire-R. Issue 0.4, JAXA, 13 August 2015.
- [4] K. Iwase et al., "The evaluation of SpaceWire-R draft specification through the connectivity test using SpaceCube2," Proc. 6th Int. SpaceWire Conf., pp. 82-85, Athens, 2014.
- [5] T. Yuasa, T. Takahashi, M. Nomachi, and H. Hihara, "A SpaceWire router architecture with non-blocking packet transfer mechanism," Proc. 6th Int. SpaceWire Conf., pp. 204-210, Athens, 2014.
- [6] T. Yuasa, H. Hihara, and T. Yamada, "SpaceWire-R core implementation and performance study," 24th SpaceWire Working Group Meeting, Noordwijk, September 2015.
- [7] W. Mich et al., "Implementation and validation of the SpaceWire-R protocol," Proc. 7th Int. SpaceWire Conf., pp. 41-44, Yokohama, 2016.
- [8] K. Romanowski and P. Tyczka, "SpaceWire-R protocol implementation: test results, assessment, and future work," 26th SpaceWire Working Group Meeting, Noordwijk, April 2017.
- [9] A. Tavoularis, V. Vlagkoulis, N. Pogkas, V. Kollias, and K. Marinis, "iSAFT-PVS: recording, simulation & traffic generation at full network load," Proc. 6th Int. SpaceWire Conf., pp. 73-79, Athens, 2014.
- [10] S. Mills et al., "Developing SpaceWire Devices with STAR-Dundee test and development equipment," Proc. 5th Int. SpaceWire Conf., pp. 257-260, Gothenburg, 2013.
- [11] K. Romanowski et al., "SpaceWire network management using Network Discovery and Configuration Protocol," Proc. 7th Int. SpaceWire Conf., pp. 45-50, Yokohama, 2016.
- [12] M. Nomachi, "SpaceWire-R interoperability test," 28th SpaceWire Working Group Meeting, Noordwijk, March 2018.
- [13] T. Takahashi et al., "Space Cube 2 an onboard computer based on Space Cube architecture," Proc. 2nd Int. SpaceWire Conf., pp. 65-68, Dundee, 2007.
- [14] T. Yamada, SpaceWire-R. Issue 0.3, JAXA, 13 September 2013.
- [15] M. Gardner et al., Joint Architecture Standard (JAS). Reliable Data Delivery Protocol (RDDP) Specification. Sandia National Laboratories, Report SAND2011-3500, May 2011.
- [16] K. R. Fall and W. R. Stevens, TCP/IP Illustrated. Volume 1: The Protocols, 2nd ed., Addison Wesley, 2012.

Test & Verification 1 (Long)

Computer-Aided Design System for Onboard SpaceWire Networks

Test and Verification, Long Paper

Yuriy Sheynin, Valentin Olenev, Irina Lavrovskaya, Ilya Korobkov, Lev Kurbanov Saint-Petersburg State University of Aerospace Instrumentation Saint Petersburg, Russia sheynin@aanet.ru, {valentin.olenev, irina.lavrovskaya, ilya.korobkov, lev.kurbanov}@guap.ru

Abstract—The paper provides an analysis of existing simulation tools for the on-board and local area networks. We overview the main abilities of the existing software and then propose the computer-aided design (CAD) system for SpaceWire onboard networks design and simulation - SANDS. SANDS system will support the full on-board network design and simulation flow, which begins from the network design and simulation flow, which begins from the network topology automated generation and finishes with getting the network structure, configuration and parameters setting, simulation results and statistics. The paper also provides use cases for SANDS application.

Index Terms—SpaceWire, CAD, SANDS, onboard network design, simulation.

I.INTRODUCTION

Evolution of microelectronics has led to the growth of the onboard networks and systems sizes. Modern on-board networks consist of a huge number of nodes with different functionality. Interconnection of these systems is done via the on-board network with numerous devices that work with data transmission speeds, transmit different types of data with different intensity.

SpaceWire technology is easy to understand, it is compact and efficient in implementation. SpaceWire became a common technology for spacecraft on-board networking. It is easy to build and operate SpaceWire networks while they are small enough, while they are used for well predictable and not complicated set of information streams without strong requirements and constraints. However, as a network becomes larger, over some dozens of nodes, information streams becomes more dense, requirements and constraints stronger to design and operate SpaceWire network becomes tough tasks.

The motivation for a research was to create a new computer-aided design system for SpaceWire networks which will support full onboard network design and simulation flow, which begins from the network topology design and finishes with getting the simulation results, statistics and different diagrams. The paper will describe SpaceWire Automated **Dmitry Dymov**

JSC "Academician M.F. Reshetnev" Information Satellite Systems" Zheleznogorsk, Russia dymovdv@mail.ru

Network Design and Simulation (SANDS) – the new CAD system for SpaceWire networks.

Also the paper will provide a short description of the tasks that should be solved in SANDS in comparison with the existing simulation tools for on-board and local area networks. We will provide a description of each CAD component with more details and features and present the graphical user interface. Finally, we will give particular SANDS use cases.

II.NETWORK SIMULATION TOOLS OVERVIEW

Network simulators allow researchers to test the scenarios that are difficult or expensive to imitate in real world. It is particularly useful to test new communication protocols or to change the existing protocols in a controlled and reproducible environment. Simulators can be used to design different network topologies using various types of nodes. There are different types of network simulators and they can be compared on the basis of the following features:

- range from very simple to very complex,
- ability to specify nodes and links between those nodes and the traffic between the nodes,
- ability to specify everything about protocols used to handle traffic in a network,
- graphical user interface allows users to easily visualize operation of their simulated environment,
- text-based applications permit more advanced forms of customization;
- programming-oriented tools providing a programming framework that customizes to create an application that simulates the networking environment to be tested [1].

There are different network simulators with different features. Some of them are commercial, which means that the source code of the software or the affiliated packages is not provided to users. All users have to pay to get a license to use this software or pay to order specific packages for their own specific usage requirements. On the other hand, open source network simulators and their interfaces are completely open for the developers.

Currently there is a number of tools and models that give an ability to simulate the operation of communication networks, but mostly these tools are intended for the Ethernet and Wi-Fi networks. Some of network simulators are overviewed in the current paper.

A. QualNet

QualNet is the commercial tool which provide implementations of layers/modules and features like GUI based analysis tools. It runs on all common platforms (Linux, Windows, Solaris, OS X) and is specialized in simulating all kind of wireless applications. It has a quite clear user interface while also offering an easy to use command line interface [2]. The QualNet user interface is shown in Fig. 1.



Fig. 1. QualNet user interface

QualNet is composed of the following components:

1) QualNet Architect: a graphical scenario design and visualization tool. In Design mode, it is possible to set up terrain, network connections, subnets, mobility patterns of wireless users, and other functional parameters of network nodes. Visualize mode gives an ability to perform in-depth visualization and analysis of a network scenario designed in Design mode. As simulations are running, users can watch packets at various layers flow through the network and view dynamic graphs of critical performance metrics.

2) *QualNet Analyzer:* a statistical graphing tool that displays hundreds of metrics collected during simulation of a network scenario.

3) QualNet Packet Tracer: a graphical tool that provides a visual representation of packet trace files generated during the simulation of a network scenario.

4) QualNet File Editor: a text editing tool.

5) *QualNet Command Line Interface:* Command line access to the simulator [3].

Therefore, QualNet has a lot of benefits and useful components, but unfortunately, it is a very expensive solution.

B. OPNET

OPNET is a registered commercial trademark and a name of product presented by OPNET Technologies incorporation. It became one of the most famous and popular commercial network simulators. Because of its use for a long time in the industry, it has become mature and has occupied a big market share. OPNET claims to be the fastest simulation engine among leading industry solutions. It has a wide variety of niche simulators for the wired/wireless areas. It also has many of wired/wireless protocol and vendor device models with source code, and allows object-oriented modeling of components. Modeling environment is an hierarchical one and has a slightly more complex method of definition of nodes as finite state machines. The simulator is flexible and, therefore, allows integration with other libraries and simulators. With help from a rich suite of integrated, GUI-based debuggers and analyzers the setup, configuration can be done. The OPNET user interface is shown in Fig. 2.



Fig. 2. OPNET user interface

OPNET inherently has three main functions: modeling, simulating, and analysis. [1], [4].

C. NS-3

NS-3 is an open source discrete-event network simulator which targets primarily for research and educational use. NS-3 is licensed under the GNU, GPLv2 license, and is available for research and development.. NS-3 uses a Python simulation core, its protocol entities are designed to be closer to real computers. Moreover, NS-3 uses lightweight virtual machines. NS-3 is developing a tracing and statistics gathering framework trying to enable customization of the output without rebuilding the simulation core [1], [5]. The NS-3 user interface is shown in Fig. 3.



Fig. 3. NS-3 user interface

D. OMNet++

OMNeT++ is a C++-based discrete event simulator for modeling communication networks, multiprocessors and other distributed or parallel systems. OMNeT++ is public-source, and can be used under the Academic Public License that makes the software free for non-profit use. OMNet++ offers an easy to use GUI for graphical network editing, animation and configuring simulation runs. OMNet++ has a basic output analyzer, which can display collected statistics in graphical formats. However, not many OSI-related models are implemented. Nevertheless, its base infrastructure is very extensible, and it is easy to modify. This offsets the lack of implemented models to a certain extent. The models or modules of OMNeT++ are assembled from reusable components as OMNeT++ is designed to provide a componentbased architecture. The OMNeT++ user interface is shown in Fig. 4.



Fig. 4. OMNeT++ user interface

OMNeT++ aims at providing a rich simulation platform, and leaves creating simulation models to independent research groups [1], [6].

E. J-Sim

J-Sim (formerly known as JavaSim) is a component-based, compositional simulation environment, implemented in Java. J-Sim is similar to OMNeT++ in that simulation models are hierarchical and built from self-contained components. J-Sim is also a dual-language simulation environment, in which classes are written in Java, and glued together using Tcl (or Java) which makes implementing graphical editors impossible. In fact, J-Sim does provide a graphical editor (gEditor), but its native format is XML. However, the problem with XML as native file format is that it is hard to read and write by humans. Simulation models are provided in the Inet package, which contains IPv4, TCP, MPLS and other protocol models.

The fact that J-Sim is Java-based has some implications. On one hand, model development and debugging can be significantly faster than C++, due to existence of excellent Java development tools. However, simulation performance is significantly weaker than with C++, and, moreover, it is not possible to reuse existing real-life protocol implementations written in C as simulation models [8]. Example of J-Sim gedit application is shown in Fig. 5.



Fig. 5. J-Sim gedit user interface

The summary of general purpose network simulation tools is given in Table I.

TABLE I. NETWORK SIMULATION TOOLS SUMMARY

Feature	QualNet	OPNET	NS-3	OMNeT++	J-Sim
Language	Parsec C++	C (C++)	C++/ Python	C++	Java, Tcl
GUI	Very good	Good	Moderate	Good	Good
Availability	Comme rcial	Commer- cial	Free	Free for non-profit use	Free
Scalability	Good	Good	Good	Good	Good
Fast simulation	Very good	Very good	Moderate	Moderate	Poor

III.SPACEWIRE NETWORK SIMULATION TOOLS

Most of considered network simulation tools are related to Ethernet and Wi-Fi networks simulation and unfortunately, none of these tools can model SpaceWire networks. However, this overview gives a good vision for the development of a tool for the SpaceWire simulation, because it shows the useful abilities and mechanisms, user-friendly GUIs and additional features for the network operation analysis. In addition, let us consider some tools that are specially developed for SpaceWire networks modeling. Most of them are based on the OPNET simulation framework. For these purposes OPNET was adapted for the SpaceWire and updated with a list of specific modules and network elements.

This way was chosen by Thales Alenia company, which implemented MOST (Modeling of SpaceWire Traffic) [7] for the European Space Agency. Initially MOST was based on the OPNET toolkit dedicated to network modeling. Currently, MOST is available for NS3 simulator.

The MOST library contains SpaceWire nodes, routers and links which are selected by the user to build the SpaceWire network topology thanks to drag & drop actions. Configuration is done at the network level thanks to a set of attributes attached to each network component that can be tuned by the user. In MOST the Building Block (BB) concept is used to identify different communication layers in the node layout and it is in order to offer to user a flexible simulation tool. For example, the "SpaceWire standard BB" (or CODEC) is a SpaceWire node interface, which ensures node physical connection to the SpaceWire network with the data transmission management. This "SpaceWire standard BB" is clearly separated from the protocol "RMAP BB". As a result, one BB can be enhanced anytime without impacting others BB. Examples of MOST capabilities are shown in Fig. 6.

OPNET offers a tool to analyse simulation output. It provides various ways to display each type of data. SpaceWire traffic analysis is done based on observation of statistic parameters such as: the end-to-end delay, the bottleneck observation, packet size, packet latency or jitter, number of sent and received packets, evaluation of the sustained bandwidth and also buffers occupation. Observables are selected by the user and it can be chosen to observe the data at the node level (internally) or at the network level [8].



Fig. 6. MOST user interface based on OPNET

Similar solution has been developed by Sandia National Laboratories (SNL) [9], but it gives less abilities than MOST, because it does not have an option to insert errors to the transmitted data, which is not good for testing and verification. The SNL team also used extension features within OPNET Modeler to create a set of general purpose modules representing different network elements or basic building blocks for SpaceWire networks simulation. The modules include models of SpaceWire nodes, routers, broadcast servers, and links. These modules can be arranged to represent networks during the design stage. Then, these networks can be analyzed for the desired behavior.

The second ability of the tool is an in-depth analysis of the accurate distribution of system time across the SpaceWire network. To accomplish this task, the SNL team developed a packet broadcast mechanism that would lay upon the standard SpaceWire protocol. A representative SpaceWire network was constructed within the OPNET Modeler simulation environment. Based on this network representation, several simulations were executed to study the behavior of the network with respect to packet transmission time, jitter, and the accuracy of distributed system time.

SpaceWire models provide a generalized tool for examining network behavior and different network designs. The user interface is very similar to MOST because of the OPNET base.

However, there is a tool that is not based on the OPNET. It is VisualSim SpaceWire modeling, developed by MIRABILIS Design Company [10]. VisualSim is intended for end-to-end system-level design. The graphical nature of the product and the availability of the parameterized library, SmartBlocks, make the tool easy to use, adopt and learn. VisualSim combines DSP, Analog, Protocols and Digital Architecture in a simulation model. SmartBlocks are graphical single representations of hardware, software and networking components at queuing, performance, transaction and cycleaccurate levels of abstraction. VisualSim can be used for performance trade-offs using metrics such as bandwidth utilization, application response time and buffer requirements. Architecture analysis of arbitration algorithms, component sizing, software instruction optimization, hardware-software trade-offs and system coverage. Therefore, VisualSim gives an ability to test the real hardware SpaceWire devices, but it is not applicable for prototyping of real onboard networks on early stages of the project. The VisualSim user interface is shown in Fig. 7.



Fig. 7. VisualSIM user interface

Consequently, there are only three tools that give an ability to build SpaceWire networks and to simulate their operation. Two of them are based on the OPNET and use its capabilities. Moreover, all these three tools are commercial and rather expensive.

The SUAI team has good experience in network modeling and building SpaceWire network models. In addition, we have a tool, which is able to simulate basic SpaceWire networks and transport-layer protocols – DCNSimulator [11].

The Digital Communication Network Simulator (DCNSimulator) is a tool for design, system-level simulation and analysis of networks. DCNSimulator is based on Qt and SystemC. It consists of the simulation engine and libraries of network components. The simulation engine is a general part that could work for simulation of any network. Libraries of network components are specific for particular network standards and could represent network components at various details levels – from general virtual components to cycle-accurate models of particular devices. Simulated device models are implemented in C++. Application software algorithms

could run at end nodes thus generating realistic traffic for the simulated network. The simulator also allows users to design networks graphically in MS Visio. The DCNSimulator runs in Windows and does not require any other third party software for its operation. The DCNSimulator user interface is shown in Fig. 8.

The DCNSimulator with its library completely supports SpaceWire networks. It implements all levels of the SpaceWire standard (excluding signal and physical ones) and provides models of a terminal node, a routing switch and a channel (parameterized point-to-point link). Network models can be composed of these models of network elements. It also supports error imitation for channels and devices. With this tool, SpaceWire networks can be analyzed at the levels of bit flows, characters and packets. Therefore, one can analyze control codes and data packets propagation, channel workload and all errors occurred in channels. The simulator displays appropriate charts, statistics and information about every transferred code and packet.

We gathered the requirements for such kind of a simulation tool from industry. Analysis of these requirements showed that current version of the DCNSimulator also does not meet all industry requirements. Therefore, we started redesigning the existing tool in order to make it more useful and convenient in application.



Fig. 8. DCNSimulator user interface

IV.SPACEWIRE AUTOMATED NETWORK DESIGN AND SIMULATION

To be useful for the needs of the large industrial companies the following abilities should to be provided by the network simulation tool:

• Simulation of network models that are implemented according to the SpaceWire and SpaceWire-RUS

standards. That means, it has to have GigaSpaceWire extension;

- Simulation of the protocols of transport and application levels in nodes (at least RMAP [12] and STP-ISS [13]);
- Simulation of fault tolerance and redundancy;
- Simulation tool should be able to build a SpaceWire network consisting of network regions;
- Simulation tool should be able to save different designed objects (like nodes or switches) into a separate library that could be used for the other projects;
- Simulation tool should be able to build and model a network consisting of up to 1024 nodes.
- Simulation tool should give an ability to provide such simulation results like traffic between any two nodes, latencies, packet delivery time, transmission errors, channel bandwidth, etc.

Therefore, we made a decision to update the DCNSimulator with a number of new features that would give an ability to use the simulation tool during the whole process of the spacecraft onboard network design. This simulation tool would be a part of a complex computer-aided design system – SpaceWire Automated Network Design and Simulation (SANDS). This system will consist of four main components, and simulation tool would be only the one of them. SANDS will be a system for SpaceWire on-board networks which will support full onboard network design and simulation flow, which begins from the network topology automated generation and finishes with getting simulation results, statistics and different diagrams. SANDS will include all the improvements and extensions that are needed for the space industry. SANDS architecture is shown in Fig. 9.

SANDS architecture includes four main components:

- Component #1: A component for onboard network topology design and evaluation of its structural characteristics;
- Component #2: A component for tracking of the routes for the data transmission in a network;
- Component #3: A component for generation of the scheduling table for the STP-ISS transport protocol for the transmission of the data with Scheduled quality of service;
- Component #4: A component for simulation of the network operation with all the data that component got from other 3 components and graphical user interface (majorly redesigned DCNSimulator) [11].

We plan to implement a flexible system, with possibility of addition new protocols. Currently, we plan to include in SANDS implementation of the SpaceWire protocol and two transport layer protocols: RMAP [12] and STP-ISS [13].



Fig. 9. Architecture of SpaceWire Automated Network Design and Simulation toolset

Visualization and graphical interface will be taken from the VIPE project [14]. Graphical user interface (GUI) will provide the visual network composition and management capabilities. It will allow designing SpaceWire network topology in visual interactive way from components (see Fig. 10).



Fig. 10. SANDS graphical user interface

The component library is a replenish set of network nodes and switches relevant to physical devices that are available for network building. It may also include flexible components if the developer wants to try various combinations of network equipment. For all nodes, switches and channels the GUI will provide the configuration interface to set up their parameters, configure transport protocols and application-level traffic generators.

The designed network will be exported to the intermediate representation format to be used in other SANDS components for simulator, routes tracking, scheduling calculation and other tools. GUI will be also able to show results after running each component.

V. SANDS FUNCTIONALITY

A. Network topology design and evaluation of its structural characteristics

SANDS provides strong network structure analysis features. Component #1 is responsible for the following tasks:

- network topology design;
- evaluation of structural characteristics of the designed topology;
- network topology transformation for achieving required fault-tolerance.

Network topology design assumes creation of a network topology by means of GUI and setting up parameters for nodes, switches and links. Once this step is completed, the designed network can be analysed. We suppose to estimate the following characteristics of the designed network:

- Network mass: cable mass, switches mass or both;
- Power consumption of network switches;
- Network diameter

• Fault-tolerance of the network or its cluster.

This functionality is already implemented in SANDS.

B. Tracking of deadlock-free routes for the data transmission in a network

Another SANDS functionality is data transmission routing in a network without deadlocks. This is one of the most important tasks in network design; it implies finding sequence of switches between source and target devices and creating routing tables for these switches. SpaceWire technology provides opportunity of creating onboard computing network with flexible network architecture and high throughput. Adjusting routing tables and adaptive routing tables in switches makes possible of using static routing in SpaceWire network.

Moreover, the routes tracking tool gives an ability to generate redundant routes in order to tolerate faults in onboard network.

In accordance with tracked routes SANDS will provide routing tables for configuration of routers in onboard network. In addition, this part of SANDS will provide estimation of worst-case data and control codes transmission latencies.

C. Generation of scheduling tables

Nowadays there are transport layer protocols for SpaceWire that provide time division multiplexing mechanisms (e.g. STP-ISS, SpaceWire-D). Scheduling tables creation is a very complex problem which is very difficult to solve for big networks with multiple data exchanges.

SANDS will provide functionality for generation of scheduling tables for STP-ISS scheduling quality of service. These scheduling tables will take into account the network structure and tracked routes.

D. Simulation of the network operation

Finally, SANDS will provide simulation features. Simulation core is based on the DCNSimulator extended with additional functionality and is implemented in SystemC. There will be two modes for simulation of network operation with different levels of details:

- Bit level simulation of full SpaceWire stack and Transport protocol or Application. This mode provide very detailed simulation which is not very fast.
- Packet level simulation of upper layers only: Network, Transport, Application. This mode omits some link level mechanisms and provide very fast simulation.

The user will be provided with detailed statistical log tracking all events of interest in the network and visual diagrams and graphics with simulation results.

VI. SANDS USE CASES

In this section we provide examples of SANDS Component#1 functionality.

A. Estimation of structural characteristics and fault tolerance

Firstly, we will provide a use case of analysis features of SANDS. Fig. 11 shows a network which consist of 12 terminal nodes (blocks in green) and 6 routers (blocks in blue and light blue).



Fig. 11. Example of network topology analysis

After performing analysis SANDS gives a detailed analysis report including mass parameters for routers and cables, power consumption estimation, minimal and maximal link rates. Moreover, it performs fault-tolerance analysis which is one of the key features of this analysis [15]. Fault-tolerance analysis for this network structure results in zero fault tolerance. There is a bottleneck in this network in its bottom part containing the following nodes and routers: N5 - N12, R5 and R6. This part of a network cannot tolerate any fault. The user is provided with all this information in the report and corresponding highlights in GUI.

B. Network topology transformation

Let us show an example of algorithm application to a network with regions (see Fig. 12). The initial network topology consists of three regions. The whole network is not fault-tolerant, however, Region 3 is *1*-fault-tolerant, because of cross-connections between nodes' units and routers R3 and R4.



Fig. 12. Example of network transformation

We ran SANDS software to obtain a network which is *l*-fault-tolerant. The result of network transformation is shown in Fig. 13. All new elements are shown in purple.



Fig. 13. Transformed topology

Our network transformation algorithm added additional redundant ports to the terminal nodes N1, N2, N3 and N4 for cross-connections. Each terminal node in Region 1 and Region 2 contains two SpaceWire ports which are connected to different routers in order to increase fault tolerance. Moreover, we can observe two new routers R1_1 and R2_1, which were added to increase fault tolerance in Region 1 and Region 2. Topology transformation tool added 5 additional communication links. Region 3 was not transformed as it was originally *1*-fault-tolerant. The resulting network topology is *1*-fault-tolerant.

VII. CONCLUSION

In current paper, we overviewed existing simulation tools for on-board and local-area networks and proposed SANDS - a new computer-aided design system for SpaceWire onboard networks design and simulation. This software will solve important tasks, which spacecraft developers face with during implementation of satellites and other space vehicles. SANDS will begin its workflow from the on-board network topology design and estimation of its physical characteristics. Then it will give an ability to track the routes for the data transmission in a network and generate scheduling tables for the STP-ISS transport protocol for data transmission with Scheduled QoS. After the network design is finished - the last stage would be simulation of the network operation with real characteristics. Graphical user interface provides possibilities to draw the network topology and set different parameters of nodes, switches and channels. The proposed software system would a good assistant during the spacecraft design, implementation and testing. Finally in the paper we provided use case for SANDS network analysis component.

The component for onboard network topology design and evaluation of its structural characteristics is already implemented and ready to use, while all other functionality is in progress and estimated to be finished by the end of 2018.

ACKNOWLEDGMENT

The research leading to these results has received funding from the Ministry of Education and Science of the Russian Federation under the contract RFMEFI57816X0214.

REFERENCES

[1] Mrs. Saba Siraj, Mr. Ajay Kumar Gupta, Mrs Rinku-Badgujar. "Network Simulation Tools Survey", *International Journal of* Advanced Research in Computer and Communication Engineering, Vol. 1, Issue 4, Wagholi, 2012, pp. 201-210

- [2] T. Doerffel, "Simulation of wireless ad-hoc sensor networks with QualNet", Advanced Seminar on Embedded Systems, Technische Universitat Chemnitz, 2009, 16 p.
- [3] SCALABLE Network Technologies, "Make Networks Work. Network modeling software for Development and Analysis", QualNet Datasheet, 2014, 4 p.
- [4] Hou Jianru, Chen Xiaomin, Sun Huixian, "An OPNET Model of SpaceWire and Validation", Proceedings of the 2012 International Conference on Electronics, Communications and Control, Zhoushan, 2012. pp. 792-795
- [5] NS-3 Manual, "NS-3 Network Simulator", 2017, 165 p.
- [6] A. Varga, R. Hornig "An overview of the OMNeT++ simulation environment", Proceedings of the 1st international conference on Simulation tools and techniques for communications, networks and systems & workshops, Marseille, France, 2008.
- [7] B. Dellandrea, B. Gouin, S. Parkes, D. Jameux, "MOST: Modeling of SpaceWire & SpaceFiber Traffic-Applications and Operations: On-Board Segment", Proceedings of the DASIA 2014 conference, Warsaw, 2014.
- [8] Thales Alenia Space, "Modeling Of SpaceWire Traffic", Project Executive Summary & Final Report, 2011, 25 p.
- [9] B. van Leeuwen, J. Eldridge, J. Leemaster, "SpaceWire Model Development Technology for Satellite Architecture", Sandia Report, Sandia National Laboratories 2011, 30 p.
- [10] Mirabilis Design, "Mirabilis VisualSim data sheet", 2003. 4 p.
- [11] A. Eganyan, E. Suvorova, Y. Sheynin, A. Khakhulin, I. Orlovsky, "DCNSimulator – Software Tool for SpaceWire Networks Simulation", Proceedings of International SpaceWire Conference 2013, 2013, pp. 216-221.
- [12] ESA. Standard ECSS-E-ST-50-52C, SpaceWire Remote memory access protocol. Noordwijk : Publications Division ESTEC, February 5, 2010.
- [13] Y. Sheynin, V. Olenev, I. Lavrovskaya, I. Korobkov, D. Dymov "STP-ISS Transport Protocol for Spacecraft On-board Networks", Proceedings of 6th International SpaceWire Conference 2014 Program; Greece, Athens, 2014. pp. 26-31.
- [14] Syschikov, A., Sheynin, Y., Sedov, B., Ivanova, V. "Domain-specific programming environment for heterogeneous multicore embedded systems", International Journal of Embedded and Real-Time Communication Systems, Volume 5, Issue 4. 2014, pp. 1-23.
- [15] Lavrovskaya I., Olenev V., Korobkov I. "Fault-Tolerance Analysis Algorithm for SpaceWire Onboard Networks" in Proceedings of the 21st Conference of Open Innovations Association FRUCT, University of Helsinki, Helsinki, Finland, 2017, pp. 217-223.

Testing and validation of SpaceWire control and data links of RC64

Test and Verification session

Ran Ginosar, Peleg Aviely, Roy Nesher, Zeev Meister, Tsvika Israeli, and Dror Reznik Ramon Chips Ltd.

Yokneam, Israel ran@ramon-chips.com

Abstract— The RC64 processor includes six SpaceWire interfaces. Two interfaces support SpaceWire Remote Memory Access Protocol (RMAP) for program load, control, and monitor. Four interfaces are software defined, requiring software for direct memory access (DMA) configuration and protocol handling. The verification and test procedures of the SpaceWire interfaces of RC64 are demonstrated in this report.

Index Terms—SpaceWire, SpaceFibre, VPX, RC64, GR712RC

I. INTRODUCTION

RC64 contains 64 digital signal processing (DSP) cores, 4MB shared on-chip memory, an interface to double data rate (DDR2/3) external memory, and 12 SpaceFibre high speed serial links. It also contains two SpaceWire links for RMAPbased control and four SpaceWire links for data. The control links enable external RMAP access to the entire memory and register address space in RC64. They are designed for booting RC64, for storing code and data in it, for reading status and fault detection, isolation, and recovery (FDIR) data, and for controlling RC64 recovery from faults. RC64 may be controlled by an external high-reliability processor such as the GR712RC dual-core LEON3FT processor, or by another RC64. In fact, the two symmetrical SpaceWire control links enable a two-way ring of RC64 chips that may or may not include one or more control processors such as GR712RC. The data links do not offer RMAP and may be used to connect to instruments, or to interconnect multiple RC64 processors in a medium data rate network. We describe how all these capabilities are tested and validated, and how they enable testing and validation of other functions of RC64.

II. TEST-BED

The test-bed, described in Fig. 1, includes two highperformance VPX64 processor boards in a 3U VPX chassis. Each board, described in Fig. 2 and pictured in Fig. 3, includes a single RC64 processor with its periphery. The LEON3 system controller, pictured in Fig. 4, connects through a SpaceWire link to the front panel interface of each VPX64 board.



Fig. 1. Test-bed design



Fig. 2. VPX64 board architecture



Fig. 3. VPX64 board

The VPX64 board includes these features:

- 3U VPX standard board
- 200MHz RC64 processor, 64 DSPs, 4Mbyte shared memory
- 8GB non-volatile flash memory
- 2GByte 666MHz DDR3 SDRAM
- Four SpaceWire links, each with 200Mbps speed per direction for data communication
- Two SpaceWire links, each with 100Mbps speed per direction for host control through RMAP protocol
- Twelve SpaceFibre links, each with up to 6Gbps speed per direction with an aggregate bandwidth of up to 120 Gbps
- Optional interface to two digital-to-analog converters (DAC) or two analog-to-digital converters (ADC) for communication or radar purposes
- JTAG debugger bus interface for debugger control
- Optional interface with LEON3 host or PC host through a SpaceWire connector

• Optional interface with PC through a SpaceFibre connector

The VPX chassis includes these features:

- 3U VPX backplane, providing reset, power and inter-board connection of SpaceWire and SpaceFibre interfaces
- Backplane support for JTAG and reset per board



Fig. 4. LEON3 GR-MCC-C

- Current and voltage meter display
- Air temperature sense and display
- Fan speed control and configuration
- SpaceFibre ring topology providing six links between each adjacent pair of boards (including between the last board and the first board for a full circle) at up to 6.25Gbps full duplex
- SpaceWire data ring topology providing two links between every adjacent pair of boards at up to 200Mbps full duplex
- Ring topology supporting a SpaceWire host link between every adjacent pair of boards at up to 100Mbps full duplex (the ring is broken if using the front panel SpaceWire interface for host purposes)

The LEON3 GR-MCC-C system controller includes these features:

- Debugger interface accessible through JTAG for the GRMON debugger
- Two high-speed SpaceWire links, allowing seamless data transmission to/from RC64 processors
- Real-time operating system (RTEMS), providing access to basic functionalities required for data transmission, communication, and software management

Peripheral equipment includes:

- Brick Mk3 for the PC-to-SpaceWire interface, used for host control functions from the PC using USB3
- Lauterbach debugger, used for RC64 debug programs, connected using the JTAG interface to the VPX64 board. Connects using Ethernet to LAN

The PC platform includes:

- Interface to Brick Mk3
- LAN interface for Lauterbach debugger box
- Software development environment for RC64
- Lauterbach debugger software
- Script environment for test execution on RC64



Fig. 5. PC-based host control

III. BRICK MK3 CONTROL

The PC-based host control test-bed (see colored boxes in Fig. 5) uses the Brick Mk3 USB3 to SpaceWire equipment to operate the VPX64 boards.

The PC includes the Phyton environment, which supports the SpaceWire RMAP protocol-based interface with the RC64 processor on the VPX64 boards. Software developed for the environment supports these functions:

- 1. Reset the VPX64 boards
- 2. Control clock generator configuration on the VPX64 boards
- 3. Load and launch executables on the RC64 processors
- 4. Read/write access to the RC64 internal address space
- 5. Monitor board temperature

Test programs for the environment demonstrate the above sequence with many types of programs that verify the VPX64 board and RC64 functionality.

IV. LEON3 CONTROL

The LEON3-based host control test-bed (see colored boxes in Fig. 6) replaces the MK3 brick and PC control over VPX64 boards with the LEON3-based GR-MCC-C board. The SpaceWire interfaces of the GR-MCC-C board are used for interfacing the VPX64 boards. Programs loaded to the LEON3 processor through JTAG interface support these functions:

- 1. Reset the VPX64 boards
- 2. Control clock generator configuration on the VPX64 boards
- 3. Load and launch executables on the RC64 processors
- 4. Read/write access to the RC64 internal address space

V. SPACEWIRE RING TOPOLOGY

The SpaceWire ring topology tests use the backplane interconnect between the two boards (Fig.7). The left-



Fig. 6. LEON3-based host control



Fig. 7. SpaceWire ring topology

hand VPX64 board executes the SpaceWire routing program with these functions:

- 1. The SpaceWire host hardware detects packets with a non-zero path address, directing the packet to the DMA interface.
- 2. The software application driver activates the DMA function to handle the incoming SpaceWire packet.
- 3. The software application removes the first path address byte from the incoming packet.
- 4. The software application sends the packet from SpaceWire Host #0 to #1, or from #1 to #0.

The PC host uses the path address to program the righthand VPX64 board with test programs and validates the execution. All SpaceWire RMAP transactions route through the left-hand VPX64 board.

I. SPACEWIRE DATA

The SpaceWire data tests use the SpaceWire ring to load programs to both boards (Fig. 8). The programs perform DMA transactions through the board-to-board SpaceWire data links. SpaceWire packets from one board are compared with expected content on the other board.

II. SUMMARY

The test-bed provides a testing environment for RC64. The PC script environment supports automatic execution of the tests to validate system stability. Each test includes pass/fail criteria; reporting the results to a managed database. Multiple data rates, link-up/down, and disconnect were tested for the SpaceWire links.



Fig. 8. SpaceWire data

Test & Verification 2 (Long)

Testing SpaceFibre Equipment and Systems

Test and Verification, Long Paper

Stephen Mudie, David Gibson, Chris McClements, Stuart Mills STAR-Dundee Ltd Dundee, Scotland, UK stephen.mudie@star-dundee.com

Thorough testing is required for successful SpaceFibre equipment and system development. This helps identify defects and provides assurance equipment operates as expected. The STAR Fire Mk3 can transmit and receive SpaceFibre traffic to stimulate and emulate SpaceFibre equipment for test purposes. In addition, it can capture and display SpaceFibre traffic, aiding debug and validation efforts. The SpaceFibre Recorder increases capture functionality further, recording significantly larger quantities of traffic over multiple SpaceFibre lanes. This paper considers how the hardware and software capabilities provided by these units may be used to test SpaceFibre equipment and systems.

Index Terms— SpaceFibre, Testing, Spacecraft Electronics, STAR Fire Mk3, SpaceFibre Recorder.

I. INTRODUCTION

SpaceFibre is an on-board network technology for spaceflight applications that runs at multi-Gbit/s. Designed to support deterministic data delivery, data packets are segmented and transmitted in frames across virtual channels (VC), where each virtual channel is constrained by a quality of service (QoS) mechanism that supports priority, bandwidth reservation and scheduling. The QoS mechanism is provided in an efficient hardware interface design along with fault detection, isolation, and recovery (FDIR) capabilities. SpaceFibre uses the same packet format as SpaceWire, meaning it is backward compatible with SpaceWire at the network level, and existing SpaceWire equipment can be interconnected to a SpaceFibre network.

During SpaceFibre system development it is important to test that the system operates as expected and to debug any issues encountered. To stimulate SpaceFibre equipment and validate the response, test equipment capable of transmitting and receiving SpaceFibre traffic is needed. To further validate traffic and debug any unexpected problems, test equipment able to capture, interpret and display SpaceFibre link traffic is required.

The STAR Fire Mk3 and SpaceFibre Recorder are two pieces of SpaceFibre test equipment that allow users to emulate, stimulate, debug and validate their SpaceFibre enabled equipment. This paper describes the capabilities of these hardware units and the accompanying software, and how Steve Parkes School of Science and Engineering University of Dundee Dundee, Scotland, UK smparkes@dundee.ac.uk

these may be used to test and debug SpaceFibre equipment and systems.

II. STAR FIRE MK3

The STAR Fire Mk3 can be used both as a SpaceFibre interface device, to transmit and receive traffic, and as a SpaceFibre link analyser. It is controlled by a host PC connected by a USB 3.0 port and interfaces with SpaceFibre equipment using two SpaceFibre EGSE ports. In addition, it has two SpaceWire ports, four external SMB triggers, two MICTOR connectors and a power connector.



Fig. 1 STAR Fire Mk3 Unit

A. SpaceFibre Interface

The STAR Fire Mk3 has two means of control for transmitting and receiving SpaceFibre data, each designed for a different purpose. Data can be transmitted from and received to a host PC or built-in hardware data generators and checkers can be used. The host PC solution allows for greater control over packet contents whilst the data generators and checkers provide real-time performance.

B. SpaceFibre Link Analyser

The STAR Fire Mk3 has 512MB of memory that can be used to capture over 44 million SpaceFibre words. It can capture SpaceFibre traffic received, transmitted or inline on a SpaceFibre link. SpaceFibre traffic is captured around a trigger event (typically a SpaceFibre word) set by the user. Once capture has completed, the SpaceFibre traffic is uploaded to the host PC and presented in multiple views displaying this in different levels of detail.

C. SpaceWire to SpaceFibre Interconnect

SpaceWire ports on the STAR Fire Mk3 allow SpaceWire equipment to be interconnected to a SpaceFibre network. This means SpaceWire equipment connected to the STAR Fire Mk3 can transmit and receive SpaceWire packets over a SpaceFibre lane virtual channel, and in doing so inherit the mass saving, QoS and FDIR benefits SpaceFibre provides.

D. Interfacing with External Equipment

External triggers on the STAR Fire Mk3 provide an interface for connecting external equipment to aid testing. Input and output triggers mean the STAR Fire Mk3 can both react to an input signal from external equipment and generate an output signal to notify external equipment of an event. To further aid testing, MICTOR connectors allow decoded SpaceFibre link characters for each SpaceFibre port to be accessed from a logic analyser.

III. SPACEFIBRE RECORDER

The SpaceFibre Recorder is currently under development and builds upon the SpaceFibre link analyser capabilities of the STAR Fire Mk3. The aim of this project is to create a piece of SpaceFibre test equipment capable of capturing large quantities of traffic (terabytes) from multiple SpaceFibre lanes, whilst reusing much of the existing SpaceFibre traffic display software provided with the STAR Fire Mk3.

As SpaceFibre operates at multi-Gbit/s, capturing SpaceFibre traffic can consume large quantities of memory in a short period of time. The large storage capacity the SpaceFibre Recorder provides, means debug and validation efforts can be more comprehensive. The ability to capture multiple SpaceFibre lanes means the SpaceFibre Recorder will be able to capture SpaceFibre traffic on different links of a network for greater overall SpaceFibre network visibility. Importantly, it also means the traffic of a SpaceFibre multi-lane link can be viewed, aiding multi-lane link debug and validation efforts.

IV. SOFTWARE

The STAR Fire Mk3 is supplied with software designed to complement the hardware capabilities. Created specifically for the STAR Fire Mk3's SpaceFibre functionality are three graphical user interface (GUI) applications called STAR Fire Controller, STAR Fire Statistics and STAR Fire Link Analyser. In addition to these, the STAR-Dundee software stack named STAR-System is also provided. Further information regarding these software modules and how they are used to test SpaceFibre equipment and systems is provided in the following sections.

V. STIMULATE AND EMULATE

Whilst developing a SpaceFibre system it is often necessary to work on one piece of SpaceFibre equipment whilst other parts of the system are unavailable. In this scenario it can be useful to simulate and/or emulate the behaviour of other parts of the SpaceFibre system. Test equipment controlled by software can simulate SpaceFibre equipment behaviour. This test equipment should be capable of transmitting and receiving SpaceFibre traffic so that it can stimulate and react to the unit under test (UUT).

A. Lane Initialisation

SpaceFibre lane initialisation must occur before any data can be transferred. The data signalling rate for a SpaceFibre lane must also be the same (+/- 0.01%) in each direction. The STAR Fire Controller GUI application can be used to set the STAR Fire Mk3 SpaceFibre lane data signalling rate (between 0.6 and 3.2 Gbit/s) to match that of the UUT and to start lane initialisation. This application allows the user to view and change the STAR Fire Mk3 SpaceFibre Interface settings. If lane initialisation is successful, the lane status information shows that the lane is active.

✓ STAR-Fire Mk3 Controller [Serial #15041-0002] <u>File</u> <u>D</u> evice <u>H</u> elp	– 🗆 ×
Device: STAR-Fire Mk3 SpFi Port: 1	
Device Properties SpFi Port VC Summary VC 0 VC 1 SpFi Lane Status Active Lane : Yes Data Signalling Rate : 2 Gbps SpFi Lane Settings Auto Start Lane Start Scrambling	VC 2 VC 3 VC 4

Fig. 2 STAR Fire Controller Lane Start

B. Quality of Service

SpaceFibre virtual channels are assigned QoS parameters that are used to determine which channel should be permitted to transmit data at any one time. Priority, bandwidth reservation and scheduling are used to do this. The STAR Fire Controller can be used to configure these for each virtual channel according to the simulation requirements.



Fig. 3 STAR Fire Controller Quality of Service Settings

C. Transmit and Receive

The STAR Fire Mk3 can transmit and receive SpaceFibre packets from a host PC using STAR-System. STAR-System is

the common software stack provided with all the latest SpaceWire and SpaceFibre router and interface devices from STAR-Dundee. It consists of drivers, C and C++ application programming interfaces (APIs), examples, GUI applications and documentation. Relevant to the STAR Fire Mk3, it provides a USB driver to interface with the host PC, and APIs and GUI applications to transmit and receive SpaceFibre packets. Packets can be transmitted and received from a host PC over two virtual channels simultaneously. This can be achieved at high speed (2.2 Gbit/s combined transmit and receive data rate in a loopback configuration) due to the USB 3.0 interface.

The STAR-System Transmit and Receive GUI applications provide the simplest SpaceFibre transmit and receive capabilities. These allow the user to send and receive a single user defined packet to confirm a SpaceFibre link is operating as expected and packet addressing is correct. The STAR-System Source and Sink GUI applications provide more advanced SpaceFibre transmit and receive capabilities. Both can be used to construct and save complex packet formats. The Source application allows the user to transmit multiple packets with a software defined time between each. The Sink application can be used to receive and check the contents of multiple packets. These display statistics such as the number of data, end of packet marker (EOP) and error end of packet marker (EEP) characters transmitted and received. The Sink application can be used to transmit SpaceFibre packets to a UUT to stimulate it, whilst the Source application can receive and check the contents of SpaceFibre packets transmitted from the UUT. The statistics displayed can confirm the SpaceFibre virtual channel is operating at the required data rate and if any packet data errors were encountered.



Fig. 4 Source and Sink GUI Applications

Where greater control is required over packet contents and transmit and receive functionality, custom code can be written using the C/C++ APIs. As SpaceFibre uses the same packet format as SpaceWire, the STAR-System transmit and receive GUI applications and APIs are consistent with those provided with existing STAR-Dundee SpaceWire interface and router devices.

D. Real-Time Emulation

Real-time performance is required for accurate hardware emulation, which is not possible using software written for a general-purpose operating system. The STAR Fire Mk3 builtin data generators and checkers provide a hardware independent solution for transmitting and receiving SpaceFibre packets. Configured using the STAR Fire Controller, the data generators transmit user defined data patterns at specific data rates. The data checkers receive SpaceFibre packets and check the contents match the user defined pattern. An error count is incremented each time a difference is detected, along with a count of the number of EEPs received. Transmitting packets without software interaction from the STAR Fire Mk3 allows the user to test their connected UUT with consistent packet timing and under maximum SpaceFibre lane utilisation conditions. Receiving and checking the SpaceFibre packets transmitted from a UUT, the STAR Fire Mk3 can confirm there are no data errors.



Fig. 5 STAR Fire Controller Data Generators and Checkers

SpaceFibre includes low latency event signalling and time distribution using broadcast messages. The STAR Fire Mk3 built-in broadcast generators and checkers provide a hardware independent solution for transmitting and receiving broadcast messages. Configured using the STAR Fire Controller, the broadcast generators can periodically transmit broadcasts with a fixed pattern. This allows the user to stimulate a UUT with low latency broadcast messages transmitted with accurate timing. The broadcast checkers receive broadcast messages and check these match a fixed pattern. A broadcast error count is incremented with each broadcast received that differs from the expected pattern.

E. Repeatable Testing

Multiple different settings mean the STAR Fire Mk3 can test connected SpaceFibre equipment in a variety of

configurations using the STAR Fire Controller. To help manage different test configurations and to save time and effort, the STAR Fire Controller settings can be saved to hard disk and later reloaded.

VI. DEBUG AND VALIDATE

During SpaceFibre equipment development, testing helps identify issues. Problems can also be encountered when two or more SpaceFibre units are integrated for the first time. Without SpaceFibre traffic visibility, debugging these issues can be difficult and time consuming. With SpaceFibre traffic visibility, the cause of problems can be determined more easily and issues resolved faster.

Validation tests ensure SpaceFibre equipment meet operational requirements. These confirm functionality and performance are as expected and can be used to demonstrate the acceptance criteria of external customers has been met. SpaceFibre traffic capture and inspection can be used to confirm a SpaceFibre system is operating as expected under different conditions.

A. SpaceFibre Link Analysis

The STAR Fire Link Analyser GUI application enables the STAR Fire Mk3 to be used as a SpaceFibre link analyser. This allows the user to unobtrusively capture SpaceFibre traffic to hardware memory before it is uploaded to the host PC where it is interpreted and displayed in multiple views.

The software can be used to set an optional trigger condition to ensure SpaceFibre traffic of interest is captured to memory. This can be useful whilst debugging if problem symptoms are known (e.g. EEP received) as these can be used to set the trigger condition, resulting in a SpaceFibre traffic capture relevant to the issue. To increase memory usage efficiency, the trigger position relative to memory and the memory size are configurable. These allow the user to specify the total capture memory available and how much of this should be used pre and post trigger. Optional filtering can further help ensure memory is used efficiently. This can minimise the quantity of memory used to capture often unnecessary SpaceFibre traffic such as Idle frames.

🔯 S [*] <u>F</u> ile	TAR-I	Fire Mk3 L ger <u>V</u> iev	ink Analy v <u>F</u> ind	/ser [1422(<u>D</u> evice	0-0001] <u>H</u> elp]		-		×			
2	₩.	Trigger (Trigger (Options Condition	L	A								
	~	Enable F	ilter Triggers			End A End B					×		
	¢ 0	<u>Start Cap</u> Force Tr Stop Cap	o ture gger oture	F7		General Settings Trigger On: Word Word	•						
Filter	out IE)LE frame	s, and IDL	E and SKI	P co	Type : SDF Properties : Matc	n Prop VC	erty Va # 2	elue •				
						Errors		Trigge	r Option:	;			×
						UC Data Checker		Memory	frigger Po	sition		100	50 %
											ОК		Cancel

Fig. 6 Filtering, Trigger Condition and Trigger Memory Position

Once traffic is captured to memory it is interpreted and displayed in views that can be used to debug issues and validate normal behaviour. The STAR Fire Link Analyser software displays SpaceFibre traffic in four different views, each of which allows the user to visualise and examine the traffic in a different way.

The captured SpaceFibre traffic can be saved to file and opened on any machine with the STAR Fire Link Analyser software installed. This can be used to keep a record of SpaceFibre traffic captures and to share problems with others should a traffic issue be identified.

B. Network View

The network view provides a high-level overview of all the captured SpaceFibre traffic. Visualising the captured traffic at such a high-level is useful for identifying trends and can help locate areas of interest. Below is a screenshot of the network view. At the top of the view is a timeline and navigation bar. This indicates the quantity of captured SpaceFibre traffic currently visible (based on the zoom level) and the time of this relative to the capture trigger. Below this, the captured traffic for each virtual channel on each SpaceFibre lane is displayed within grey horizontal traffic bars. Bi-directional virtual channels are grouped into virtual networks as shown in the left side of the view. Individual and/or groups of SpaceFibre traffic items are represented as coloured bars within the traffic bars. As the user zooms in and out, the quantity of traffic on display changes along with the level of detail. Hovering over the data shown presents additional information for that data chunk e.g. number of frames represented and the capture start and end time.



Fig. 7 Network View

If debugging an issue that occurs repeatedly, the high-level overview of captured SpaceFibre traffic provided by the network view can help quickly identify how often this occurs or if it is seemingly random. It can illustrate if the issue is limited to one or more VCs or if it affects the entire SpaceFibre lane, and it will show how data and broadcast transmission surrounding errors, and the traffic as a whole are affected. For example, identifying how often issues resulting in a SpaceFibre link disconnection occur may be obvious as the traffic overview might show regular periods of no data transmission within a normal data stream.

Whilst validating SpaceFibre traffic, the network view helps identify SpaceFibre traffic patterns and any issues that might otherwise not be obvious when viewing traffic at other lower more detailed levels. Communication between two SpaceFibre nodes will often follow a pattern. For example, a processor may regularly send commands to an instrument to read its acquired data, to which the instrument responds with a specific number of packets of a defined length. The command and response pattern would be obvious in the network view. If a pattern unexpectedly differs in some way, this may highlight an area that should be investigated. The network view provides a way of quickly scanning an entire SpaceFibre traffic capture for errors or unexpected patterns. If an issue is identified, as all views are synchronised, selecting the area of interest will highlight the same section in the other views so that this can be investigated in more detail.

C. Packet View

The packet view displays the SpaceFibre packets captured. Packet timing, size and contents are displayed for each virtual channel. This view allows the user to focus on SpaceFibre packets without having to consider the SpaceFibre symbols, words and frames used to construct these. Below is a screenshot of the packet view. One half of the view displays the SpaceFibre traffic travelling in one direction whilst the other half displays the opposite direction. A central grey column is used to separate these. The left most column displays the packet timing relative to the capture trigger. Double clicking a packet expands it, displaying its cargo.



Fig. 8 Packet View

If SpaceFibre packets containing incorrect data are received or are missing, the packet view can help debug this. If the STAR Fire Mk3 is being used to capture the SpaceFibre traffic transmitted between two units (inline), it can help determine if the problem is with the transmitter or receiver i.e. are the SpaceFibre packets transmitted (those captured and displayed by the STAR Fire Mk3) the same as those received? If any packet ends in an EEP, examining the contents and timing of this packet and those surrounding it may help identify the cause e.g. timing information could suggest this packet became stuck in a routing switch and timed out.

The packet view can be used to inspect and validate the timing, length and contents of packets. Remote memory access protocol (RMAP) [1] analysis can further help validate packet data. This interprets and displays the different RMAP fields for each packet, making it easier for the user to confirm these are as expected. Built-in search capabilities help to navigate through the traffic quickly and to validate packet contents. Using this, the user can check for the presence of any packets unexpectedly ending in an EEP or to search packets for a pattern of data bytes that should be present.

D. Frame View

SpaceFibre uses frames to manage the flow of information over a SpaceFibre link. There are three frame types: data, broadcast and idle frames. Data frames are transmitted across a SpaceFibre link over virtual channels whilst broadcasts are transmitted over a broadcast channel. Idle frames are transmitted when there is no data or broadcast frames to be transmitted. Virtual channels provide multiple independent communication channels over a single physical link.

The frame view displays interleaving data and broadcast frames in their appropriate channel. This allows the user to focus on the segmented SpaceFibre packet data frames and broadcast messages without having to consider the SpaceFibre symbols and words used to construct these. Below is a screenshot of the frame view. As with the packet view, the left most column displays the time relative to the capture trigger. Every other column represents a virtual channel or broadcast channel as indicated by the column header. The traffic travelling in each direction is separated by a grey column.



Fig. 9 Frame View

The number of data frames shown in each virtual channel allows the user to study the impact of the virtual channel QoS settings. This may highlight one or more virtual channels are being allocated too much or too little bandwidth credit and as a result the QoS settings should be adjusted for these.

The frame view can be used to validate the timing and length of broadcast and data frames. For example, where broadcast timing is critical, such as time distribution, the time of each captured broadcast frame can help validate this performs as expected.

E. Symbol View

SpaceFibre uses 8B10B encoding to transfer 10-bit symbols over a SpaceFibre lane. A symbol can be either a control or data symbol. A group of four consecutive symbols form a data word or control word.

The symbol view displays the captured SpaceFibre symbols and corresponding words travelling in both directions over a SpaceFibre lane. Below is a screenshot of the symbol view. Similar to the packet and frame views, one half of the symbol view displays the SpaceFibre traffic travelling in one direction whilst the other half displays the opposite direction. The left most column displays the time at which each word was captured relative to the capture trigger. Each SpaceFibre lane direction is represented by five columns: four display the captured SpaceFibre symbols whilst the fifth shows the SpaceFibre word these equate to.

	Symbol 1	Symbol 2	Symbol 3	Symbol 4	End A Word	End B Word	Symbol 1	Symbol 2	Symbol 3	Symbol 4
2016.620 µs	0x25	0 xDC	0x82	0xBC	PRBS	PRBS	0x68	0x0E	0x30	0x8F
2016.640 µs	0x3E	0x75	0x63	0x37	PRBS	SDF (VC 1)	Comma	SDF	0x01	0x00
2016.660 µs	0x42	0x43	0x04	0xAD	PRBS	DATA (0x00000037)	0x37	0x00	0x00	0x00
2016.680 µs	0x1C	0x17	0x36	0x04	PRBS	DATA (0x00000038)	0x38	0x00	0x00	0x00
2016.700 µs	Conna	SDF	0x01	0x00	SDF (VC 1)	DATA (0x00000039)	0x39	0x00	0x00	0x00
2016.720 µs	OxAE	0x00	0x00	0x00	DATA (0x000000AE)	DATA (0x0000003A)	0x3A	0x00	0x00	0x00
2016.740 µs	0xAF	0x00	0x00	0x00	DATA (0x000000AF)	DATA (0x0000003B)	0x3B	0x00	0x00	0x00
2016.760 µs	0xB0	0x00	0x00	0x00	DATA (0x000000B0)	DATA (0x0000003C)	0x3C	0x00	0x00	0x00
2016.780 µs	0xB1	0x00	0x00	0x00	DATA (0x000000B1)	EDF (SEQ +48, CR	EDF	0x30	0xB2	0x75
2016.800 µs	0xB2	0x00	0x00	0x00	DATA (0x000000B2)	SIF (SEQ +48, CR	Comma	SIF	0x30	0x60
2016.820 µs	0xB3	0x00	0x00	0x00	DATA (0x000000B3)	PRBS	0x5C	0x32	0xA0	0x36
2016.840 µs	0xB4	0x00	0x00	0x00	DATA (0x000000B4)	PRBS	0xAB	0xBA	0x81	0x9B
2016.860 µs	0xB5	0x00	0x00	0x00	DATA (0x000000B5)	PRBS	0x95	0xDD	0x12	0x3D
2016.880 µs	0xB6	0x00	0x00	0x00	DATA (0x000000B6)	PRBS	0x89	0x00	0x94	0xB2
2016.900 µs	0xB7	0x00	0x00	0x00	DATA (0x000000B7)	PRBS	0x7B	0xA2	0x00	0xB9
2016.920 us	0xB8	0x00	0x00	0x00	DATA (0x000000B8)	PRBS	0x78	0xE0	0x73	0x3D

Fig. 10 Symbol View

Studying the SpaceFibre symbols and words shown in the symbol view can help identify any problems with these. Invalid symbols and words may indicate 8B10B encoding issues. Lots of unexpected negative acknowledgement (NACK) words, resulting in retransmission of frames, may help explain suboptimal performance. Examining the number of FCTs shown may identify problems with flow control.

The symbol view can be used to validate normal SpaceFibre behaviour such as lane initialisation, frame acknowledgement, frame precedence and error recovery. Using the symbol view search, all captured traffic can be checked for the absence of any SpaceFibre words that may otherwise indicate an unexpected problem e.g. NACK, lost signal or receive error word.

F. STAR Fire Statistics

In addition to the STAR Fire Link Analyser software, the STAR Fire Mk3 is also supplied with a standalone STAR Fire Statistics GUI application. This displays virtual channel and broadcast channel statistics associated with the STAR Fire SpaceFibre data/broadcast generators and checkers. A running count of data errors, EEPs and broadcast errors is displayed alongside the data generator rate, bandwidth reservation, receive data rate and transmit data rate for each channel. Virtual channel receive and transmit lane utilisation is also graphed over time.



Fig. 11 STAR Fire Statistics

Useful for debug and validation efforts, this immediately alerts the user to any errors detected by the data and broadcast checkers. The transmit and receive lane utilisation live statistics are particularly useful for visualising the effects of virtual channel QoS and data rate changes.

VII. CONCLUSION

Thorough testing is required for successful SpaceFibre equipment development. Testing helps identify defects and confirms when they are resolved. When a development is nearing completion, testing provides assurance the final equipment is reliable and performs as expected.

This paper described how the STAR Fire Mk3 unit can be used to stimulate, emulate, debug and validate SpaceFibre equipment and systems, aiding development and helping to provide assurance that equipment is operating correctly. An overview of the STAR Fire Mk3 hardware described the interfaces available and the key functionality it provides, namely, the ability to act as a SpaceFibre interface and/or link analyser, to interconnect SpaceWire with SpaceFibre, and to interface with external equipment. The SpaceFibre Recorder currently under development was also briefly described. Capable of recording very large quantities of traffic from multiple SpaceFibre lanes, this shall extend overall network visibility.

To perform SpaceFibre equipment stimulation and emulation, the capabilities of the STAR Fire Controller GUI application and STAR-System software stack were described. This includes the ability to initialise a SpaceFibre lane, set the data signalling rate, configure the quality of service parameters, transmit and receive SpaceFibre packets from a host PC, and transmit and receive SpaceFibre packets and broadcast messages directly from hardware. To aid with SpaceFibre debug and validation, the capabilities of the STAR Fire Link Analyser and STAR Fire Statistics GUI applications were described. The different live statistics presented by the STAR Fire Statistics application were listed along with some of the reasons why these are useful. Operation of the SpaceFibre Link Analyser software was described, along with the different views in which it displays captured traffic and how each of these can help debug and validate SpaceFibre equipment.

The descriptions of the hardware and software capabilities described in this paper, along with how these can help during testing, may aid those responsible for designing and implementing SpaceFibre equipment and systems in their test efforts.

REFERENCES

- [1] SpaceWire Remote Memory Access Protocol, ECSS-E-ST-50-52C, February 2010
- [2] SpaceFibre Very High-Speed Serial Link, ECSS-E-ST-50-11C DFR1, January 2018
- [3] S. Parkes, https://www.star-dundee.com/spacefibre-users-guide, SpaceFibre User's Guide, STAR-Dundee Website
- [4] S. Mudie, C. McClements, D. Gibson, STAR Fire Mk3 User Manual Rev 1.02

SpaceWire Link Analyser Mk3 and SpaceWire Recorder

Test and Verification, Long Paper

Stephen Mudie, David Gibson, Chris McClements, Stuart Mills STAR-Dundee Ltd Dundee, Scotland, UK stephen.mudie@star-dundee.com

For those responsible for testing SpaceWire equipment it is essential to be able to capture and view the traffic on a SpaceWire link in order to help validate the link is operating correctly and debug the link should any unexpected behavior be observed. The SpaceWire Link Analyser Mk3 and SpaceWire Recorder are two items of test equipment capable of transparently capturing large quantities of SpaceWire traffic on up to four links. New software written for these units displays the captured traffic in a variety of views that allow it to be examined in different levels of detail. This paper describes the hardware and software capabilities of the SpaceWire Link Analyser Mk3 and SpaceWire Recorder and provides descriptions and examples of some of the ways these can be used to perform SpaceWire link analysis.

Index Terms— SpaceWire, Testing, Spacecraft Electronics, SpaceWire Link Analyser Mk3, SpaceWire Recorder.

I. INTRODUCTION

Whilst developing and testing SpaceWire equipment, traffic visibility is essential. This allows the engineer to confirm their unit is operating correctly and aids debugging when problems are detected. Without traffic visibility there is an increased risk that issues may go undetected and problems may take significantly longer to resolve. Common examples include issues with signalling, problems with link initialisation, over and under credit allocation, disconnect errors and missing or unexpected data. To detect problems such as these, a SpaceWire Link Analyser Mk3 or SpaceWire Recorder can be used. Each of these units allow the user to unobtrusively record the SpaceWire traffic travelling in both directions over at least one SpaceWire link, providing the traffic visibility essential for link validation and debugging.

This paper describes in detail the capabilities of the SpaceWire Link Analyser Mk3 and SpaceWire Recorder, including the improvements to the SpaceWire Link Analyser Mk3 hardware and the latest generation of SpaceWire link analysis software. Key requirements identified to improve SpaceWire traffic visibility are outlined along with some of the challenges these present. The SpaceWire Link Analyser Mk3 and SpaceWire Recorder hardware, along with how these

Steve Parkes School of Science and Engineering University of Dundee Dundee, Scotland, UK smparkes@dundee.ac.uk

operate, are briefly described. An overview of the software applications used to control these is provided. Finally, the different views in which captured SpaceWire traffic is displayed and why these are well suited for SpaceWire link analysis are described, along with some examples.

II. Aims

The SpaceWire Link Analyser Mk3 and SpaceWire Recorder are both designed to enable users to unobtrusively capture and display SpaceWire traffic. In aiming to improve the traffic visibility provided by these, the following requirements were identified:

- The SpaceWire Link Analyser Mk3 memory shall be increased to capture a greater quantity of SpaceWire traffic so that debug and validation efforts can be more comprehensive.
- The SpaceWire Link Analyser Mk3 shall interface to the host PC using USB 3.0 so that large quantities of captured traffic can quickly be uploaded.
- New common software shall be written to control the SpaceWire Link Analyser Mk3 and SpaceWire Recorder. This shall be capable of displaying and navigating large quantities of SpaceWire traffic and shall be designed to support multiple device types.

The requirements above present hardware and software challenges. As the quantity of SpaceWire traffic is increased so too is the time it takes to upload this from hardware. Once uploaded, software must be capable of quickly interpreting the captured traffic and displaying this in a variety of useful ways that can be navigated easily, all whilst minimising the use of machine resources.

III. SPACEWIRE LINK ANALYSER MK3

The SpaceWire Link Analyser Mk3 transparently captures in detail the bi-directional SpaceWire traffic travelling over a single SpaceWire link. The timing information of each SpaceWire character and error is captured along with a trace of the data and strobe signals. 512MB of on-board memory is available for capturing SpaceWire traffic, meaning up to 67 million events can be recorded. The SpaceWire Link Analyser Mk3 is controlled by a host PC. Connected using a high-speed USB 3.0 interface, able to operate at 5 Gbits/s, this ensures the entire memory contents can quickly be uploaded. It interfaces with SpaceWire equipment using two SpaceWire ports and has four external SMB triggers, two MICTOR connectors and a power connector.



Fig. 1 SpaceWire Link Analyser Mk3

A. Operation

The SpaceWire Link Analyser Mk3 is designed to monitor the traffic on a SpaceWire link either between two SpaceWire link interfaces or a single interface (in loopback mode). Below is an example configuration.



Fig. 2 Capturing One SpaceWire Link

Once connected, the capture memory properties are set. These include the quantity of memory to use, memory usage relative to capture trigger and the trigger delay time. Next the capture trigger is configured, for example trigger on an error end of packet marker (EEP). SpaceWire traffic capture is then started, causing the device to continuously capture to a circular buffer whilst waiting for the capture trigger to occur. When the trigger is detected or is forced by the user, the device memory is filled. The captured SpaceWire traffic is then uploaded to the host PC where it is displayed in different views.

Should users only be interested in capturing specific SpaceWire character types, these can be enabled/disabled prior to starting SpaceWire traffic capture, improving the efficiency with which memory is used. For example, disabling Null capture can save a significant amount of memory that can instead be used to capture other character types, such as data characters.

B. Interfacing with External Equipment

External SMB triggers on the SpaceWire Link Analyser Mk3 provide interfaces for connecting external equipment to aid testing. Equipment connected to an external input trigger can cause the capture trigger to occur when an input signal is detected. This means SpaceWire traffic capture can happen when an event of interest external to the SpaceWire Link Analyser Mk3 hardware occurs. Inversely, the SpaceWire Link Analyser Mk3 can generate an external output signal when an event of interest occurs. This can be used to notify other equipment that the capture trigger has occurred or to help keep track of the timing at which specific character types are detected.

A logic analyser can also be connected to the SpaceWire Link Analyser Mk3 using a MICTOR connector. This provides access to decoded SpaceWire character information for monitoring purposes.

IV. SPACEWIRE RECORDER

The SpaceWire Recorder transparently records the bidirectional SpaceWire traffic of up to four SpaceWire links. The detail with which SpaceWire traffic is recorded is less than that of the SpaceWire Link Analyser Mk3, however the quantity is significantly greater. It can be used to record data, time-codes and link errors at high speed on many links over long periods of time. The SpaceWire Recorder is capable of recording to hard disk at an aggregate data rate of 600 Mbit/s. The maximum amount of data that can be recorded is only limited by the size of the solid-state disks (SSDs) used (1 TB as standard).

The SpaceWire Recorder is a standalone PC. It consists of a CompactPCI (cPCI) rack containing a power supply, one solidstate disk carrier, a processor board and the STAR-Dundee SpaceWire Recorder cPCI card. The SpaceWire Recorder cPCI card interfaces with SpaceWire equipment using eight SpaceWire ports. It has four external SMB triggers, two push buttons and two switches.



Fig. 3 SpaceWire Recorder

A. Operation

The SpaceWire Recorder unobtrusively records data, timecodes and link errors on up to four bi-directional SpaceWire links. Below is an example configuration.



Fig. 4 Recording Four SpaceWire Links

Once connected, the recording properties are set. Recording can be configured to stop automatically when a specific amount of data is captured or when a pre-defined period of time has elapsed relative to the recording start time. SpaceWire recording is then started and will end when the user manually stops recording, when the recording disk is full or when the data quantity or elapsed time previously specified is met. Once recording is complete, it is indexed for fast access before it is displayed.

V. SPACEWIRE TRAFFIC VIEWER SOFTWARE

New software has been written for the SpaceWire Link Analyser Mk3 and SpaceWire Recorder. A library of traffic analysis views has been created so that a consistent set of displays can be presented for all STAR-Dundee SpaceWire and SpaceFibre link analysis devices. The views presented depend upon the level of detail available from the captured traffic. The large quantities of SpaceWire traffic recorded by the SpaceWire Recorder are displayed in network and packet views. The smaller quantity, yet more detailed, SpaceWire traffic captured by the SpaceWire Link Analyser Mk3 is shown in network, packet, character and bit-stream views. By displaying SpaceWire traffic in a variety of views, different types of issues can be more easily identified and debugged. Each view is synchronised by time meaning users can select data in one view and the others will automatically display the corresponding traffic.

The views are docked within a device-specific application window that is used to configure recording properties and start/stop capture. Where possible, this is also kept consistent across devices. Where device functionality differs, custom configuration options are presented.



Fig. 5 SpaceWire Link Analyser Mk3 Application Window

The new software improves on functionality, reusability and performance compared to previous software. It has been designed to support large quantities of SpaceWire traffic and is not tied to the capture data model of a specific device. This means it performs well when displaying vast quantities of SpaceWire traffic and can be reused with different device types capable of capturing SpaceWire traffic. The following sections describe some of the functionality and the views provided in the new software, and why these are useful for SpaceWire link analysis.

VI. TRIGGERING

An optional capture trigger can be configured for both the SpaceWire Link Analyser Mk3 and SpaceWire Recorder to help capture SpaceWire traffic of interest. The capture trigger can be set to occur when an event is detected on a SpaceWire receiver or an external input signal is detected. For example, if a user wanted to debug a problem where a disconnect error occurred occasionally in one direction on a SpaceWire link, they could set the capture trigger to occur when this was detected. The captured SpaceWire traffic could then help to identify the cause of the disconnect error.

	Enabled		Number Of				
1	\checkmark	End A	•	Disconnect Error	•	1	-
2		End A	v	Time-Code Comparator	~	1	* *
3		End A	v	Time-Code Comparator	Y	1	* *
4		End A	v	Time-Code Comparator	v	1	*
5		End A	v	Time-Code Comparator	Y	1	* *
6		End A	v	Time-Code Comparator	~	1	* *
7		End A	v	Time-Code Comparator	Y	1	* *
в		End A	~	Time-Code Comparator	~	1	A

Fig. 6 Trigger on Disconnect

Sequences of events can be specified for more complex trigger conditions.

VII. NETWORK VIEW

The network view provides a high-level overview of all the recorded SpaceWire traffic. Visualising this at such a high-

level is useful for identifying trends and can help locate areas of interest. Below is a screenshot of the network view. At the top of the network view is a timeline and navigation bar. This indicates the quantity of captured SpaceWire traffic currently visible (based on the zoom level) and the time of this relative to the capture trigger. Below this, the captured traffic in each direction on a SpaceWire link is displayed within grey horizontal traffic bars. The captured SpaceWire link and direction information are shown on the left side of the view. Individual and/or groups of SpaceWire traffic items are represented as coloured bars within the traffic bars. As the user zooms in and out, the quantity of traffic on display changes along with the level of detail. Hovering over the data shown presents additional information for that data chunk e.g. number of packets represented and the packet capture start and end time.



Fig. 7 Network View

Without the network view, navigating such large quantities of traffic could be difficult as the other views present the traffic in much more detail and show only a small portion at any one time. With the network view, large quantities of traffic can be viewed, allowing the user to quickly inspect and navigate this. When used with the SpaceWire Recorder, the network view displays the traffic of multiple SpaceWire links, providing an overview of an entire SpaceWire network or at least part of it. This can help users to track the flow of packets and time-codes through a network, which can help support network debug and validation efforts. For example, it could be used to identify routing issues such as incorrect addressing, assess if a SpaceWire link is being under or over used, or to validate SpaceWire link redundancy works as expected.

VIII. PACKET VIEW

The packet view displays the SpaceWire packets recorded. Packet timing, size and contents are displayed for each direction of a SpaceWire link. This view allows the user to focus on the captured SpaceWire packets and the normal characters (data characters, end of packet markers and error end of packet markers) used to construct these, without having to consider link characters and time-codes. Below is a screenshot of the packet view. The left most column displays packet timing relative to the capture trigger time. Next to this, a time delta column displays the time difference between the time from trigger rows. SpaceWire packets travelling in each direction are shown in end columns. Each packet displayed consists of a header byte (yellow), cargo (blue) and end of packet (pink). The delta time between each row used to represent a packet is shown in the end delta column. Users can double click the cargo of a packet to expand and view the contents.

Packet View						×
	Time Delta	End A	End A Delta	End B	End B Delta	^
55.001 ms	1411.360 µs	Header: 58	4999.410 µs			
55.001 ms	100.000 ns	Cargo Size: 7 bytes	100.000 ns			
55.002 ms	640.000 ns	EOP	640.000 ns			
56.552 ms	1549.700 µs			Header: F4	2961.800 µs	
56.552 ms	100.000 ns			Cargo Size: 20383 bytes	100.000 ns	
58.590 ms	2038.290 µs			EOP	2038.290 µs	
60.001 ms	1411.390 µs	Header: 60	4999.480 µs			
60.001 ms	100.000 ns	Cargo Size: 7 bytes	100.000 ns			
60.002 ms	640.000 ns	EOP	640.000 ns			
61.552 ms	1549.670 µs			Header: 2C	2961.800 µs	
61.552 ms	100.000 ns			9F 75 98 39 27 85 E5 B4	100.000 ns	
61.553 ms	800.000 ns			14 72 59 15 15 2F BC 2C	800.000 ns	~

Fig. 8 Packet View

By inspecting packets, users can confirm these are as expected and identify issues. Address bytes can be checked, packet cargo size and contents examined, and packets ending in an EEP rather than EOP investigated further.

The packet view also has a protocol mode that can be used to interpret the cargo of a packet according to a protocol. This currently supports the Remote Memory Access Protocol (RMAP) [1]. In this mode, the fields of RMAP packets are interpreted and displayed. RMAP protocol analysis avoids the user having to manually interpret packet cargo and allows them to more easily inspect RMAP commands and replies used to transfer data.

	Time Delta	End A	End A Delta	End B	End B Delta	^
0.000 ns		Header: RMAP Command				
0.000 ns	0.000 ns	Target Address: FE				
400.000 ns	400.000 ns	Command: Read Command	400.000 ns			
400.000 ns	0.000 ns	Data Not Verified				
400.000 ns	0.000 ns	Acknowledge				
400.000 ns	0.000 ns	Incrementing Address				
600.000 ns	200.000 ns	Key: 20	200.000 ns			
800.000 ns	200.000 ns	Initiator Address: FE	200.000 ns			
1000.000 ns	200.000 ns	Transaction ID: 00A1	200.000 ns			
1400.000 ns	400.000 ns	Extended Address: 00	400.000 ns			
1600.000 ns	200.000 ns	Address: 0000003E	200.000 ns			
2400.000 ns	800.000 ns	Data Length: 000008	800.000 ns			
3000.000 ns	600.000 ns	Header CRC: FF	600.000 ns			
3080.000 ns	80.000 ns	EOP	80.000 ns			
3710.000 ns	630.000 ns			Header: RMAP Reply		
3710.000 ns	0.000 ns			Initiator Address: FE		
3810.000 ns	100.000 ns			Command: Read Reply	100.000 ns	
3810.000 ns	0.000 ns			Data Not Verified		

Fig. 9 Protocol Analysis

To help locate specific packets, the packet view has search capabilities. Users can find the next/previous packet containing a sequence of data bytes. In addition, users can search for RMAP packets containing specific fields e.g. an RMAP write command with a specific extended address and memory address. Having search capabilities such as this is important to help quickly locate data of interest and is essential when working with the large quantities of traffic that the SpaceWire Link Analyser Mk3 and SpaceWire Recorder capture.

IX. CHARACTER VIEW

Specific to the SpaceWire Link Analyser Mk3, the character view displays captured data characters, control characters, control codes and errors. It is used to visualise and debug captured traffic at the SpaceWire character and exchange levels. The timing of each event captured relative to the trigger is displayed, allowing the user to examine when events occurred and calculate the time between these. Below is a screenshot of the character view. Similar to the packet view, the left most column displays the character event time relative to the capture trigger time, followed by a time delta column. Information for each direction of SpaceWire traffic is then represented by three columns. The event column displays the data characters, control characters and control codes captured. The error column displays any errors captured. The delta column shows the time difference between two consecutive events.

	Time Delta	End A Event	End A Error	End A Delta	End B Event	End B Error	End B Delta
60.001 ms	80.000 ns	NCHAR [60]		100.000 ns	NULL		80.000 ns
60.001 ms	80.000 ns				NULL		80.000 ns
50.001 ms	20.000 ns	NCHAR [61]		100.000 ns			
60.002 ms	60.000 ns				NULL		80.000 ns
60.002 ms	40.000 ns	NCHAR [62]		100.000 ns			
60.002 ms	40.000 ns				NULL		80.000 ns
60.002 ms	60.000 ns	NCHAR [63]		100.000 ns			
60.002 ms	20.000 ns				NULL		80.000 ns
60.002 ms	80.000 ns	NCHAR [64]		100.000 ns	NULL		80.000 ns
60.002 ms	80.000 ns				NULL		80.000 ns
60.002 ms	20.000 ns	NCHAR [65]		100.000 ns			
60.002 ms	60.000 ns				NULL		80.000 ns
60.002 ms	40.000 ns	NCHAR [66]		100.000 ns			
60.002 ms	40.000 ns				NULL		80.000 ns

Fig. 10 Character View

To quickly locate SpaceWire characters, control codes and errors, the character view can be searched. Users can confirm the absence of any errors and navigate to events of interest using this.

The traffic and timing information shown in the character view can be used to validate correct behaviour. For example, in some SpaceWire networks time-codes are periodically transmitted to indicate time-slots in which transactions can be scheduled. The character view can be used to inspect and validate time-code and data character timing. Viewing the SpaceWire traffic travelling in both directions, users can examine the information exchanged between two ends of a SpaceWire link as described in the SpaceWire exchange level. Link initialisation, flow control and error handling are illustrated below.

A. Link Initialisation

SpaceWire packets and time-codes can only be transmitted and received once SpaceWire link initialisation has taken place. This is handled by the SpaceWire interface state machine. Null control codes and flow control tokens (FCTs) must be exchanged before each SpaceWire interface reaches the run state. Below is a screenshot of the character view showing the link initialisation Null/FCT handshake.

	Time Delta	End A Event	End A Error	End A Delta	End B Event	End B Error	End B Delta	^
-4530.000 ns		GOT_BIT						
-2710.000 ns	1820.000 ns	NULL		1820.000 ns				
-2130.000 ns	580.000 ns				GOT_BIT			
-1910.000 ns	220.000 ns	NULL		800.000 ns				
-1110.000 ns	800.000 ns	NULL		800.000 ns				
-310.000 ns	800.000 ns	NULL		800.000 ns	NULL		1820.000 ns	
-120.000 ns	190.000 ns	NULL		190.000 ns				
-40.000 ns	80.000 ns	NULL		80.000 ns				
0.000 ns	40.000 ns	FCT		40.000 ns				
80.000 ns	80.000 ns	NULL		80.000 ns				
90.000 ns	10.000 ns				FCT		400.000 ns	
160.000 ns	70.000 ns	NULL		80.000 ns				
170.000 ns	10.000 ns				NULL		80.000 ns	
200.000 ns	30.000 ns	FCT		40.000 ns				
210.000 ns	10.000 ns				FCT		40.000 ns	Ϊ.

Fig. 11 Link Initialisation

B. Link Flow Control

Normal characters can only be transmitted if there is space for them in the receive buffer at the other end of the link. The receive buffer indicates that there is space for eight more normal characters to the transmitter at the other end of the link by transmitting a flow control token. Below is a screenshot of the character view showing data characters transmitted after an FCT is received.

	Time Delta	End A Event	End A Error	End A Delta	End B Event	End B Error	End B Delta	1
1251.595 ms	40.000 ns	FCT		40.000 ns				
1251.595 ms	10.000 ns				NCHAR [A9]		100.000 ns	
1251.595 ms	70.000 ns	NULL		80.000 ns				
1251.595 ms	30.000 ns				NCHAR [4F]		100.000 ns	
1251.595 ms	50.000 ns	NULL		80.000 ns				
1251.595 ms	50.000 ns				NCHAR [A8]		100.000 ns	
1251.595 ms	30.000 ns	NULL		80.000 ns				
1251.595 ms	70.000 ns				NCHAR [DA]		100.000 ns	
1251.595 ms	10.000 ns	NULL		80.000 ns				
1251.595 ms	80.000 ns	NULL		80.000 ns				
1251.595 ms	10.000 ns				NCHAR [9D]		100.000 ns	
1251.595 ms	70.000 ns	NULL		80.000 ns				
1251.595 ms	30.000 ns				NCHAR [81]		100.000 ns	
1251.595 ms	50.000 ns	NULL		80.000 ns				
1251.595 ms	50.000 ns				NCHAR [77]		100.000 ns	
1251.595 ms	30.000 ns	NULL		80.000 ns				
1251.595 ms	70.000 ns				NCHAR [7E]		100.000 ns	

Fig. 12 Link Flow Control

C. Link Error Handling

When an error is detected on one end of a SpaceWire link that end disables its transmitter. This causes the other end to disconnect and also stop transmitting. After an exchange of silence, the two ends perform the Null/FCT handshake shown in the link initialisation section above. The character view screenshot below shows an escape error being detected on end A resulting in the transmitter of that end being disabled (indicated by the absence of Nulls on end B).

Character View	Character View ×							
	Time Delta	End A Event	End A Error	End A Delta	End B Event	End B Error	End B Delta	^
-50.000 ns	30.000 ns				NULL		40.000 ns	
-40.000 ns	10.000 ns	NULL		40.000 ns				
-10.000 ns	30.000 ns				NULL		40.000 ns	
0.000 ns	10.000 ns		ESCAPE ESCAPE ERROR	40.000 ns				
30.000 ns	30.000 ns				NULL		40.000 ns	
40.000 ns	10.000 ns	NULL		40.000 ns				
70.000 ns	30.000 ns				NULL		40.000 ns	
80.000 ns	10.000 ns	NULL		40.000 ns				
110.000 ns	30.000 ns				NULL		40.000 ns	
120.000 ns	10.000 ns	NULL		40.000 ns				
160.000 ns	40.000 ns	NULL		40.000 ns				
200.000 ns	40.000 ns	NULL		40.000 ns				
240.000 ns	40.000 ns	NULL		40.000 ns				
280.000 ns	40.000 ns	NULL		40.000 ns				
320.000 ns	40.000 ns	NULL		40.000 ns				
360.000 ns	40.000 ns	NULL		40.000 ns				
400.000 ns	40.000 ns	NULL		40.000 ns				~

Fig. 13 Transmitter Disabled After Escape Error Received

X. BIT-STREAM VIEW

In addition to character information, the SpaceWire Link Analyser Mk3 captures the bit-stream on a SpaceWire link. This is presented in the bit-stream view, where the data and strobe signals are graphed for each direction. This is useful for visualising data encoding. Below is a screenshot of the bitstream view. Similar to the network view, at the top is a timeline and navigation bar. A green rectangle overlay indicates the quantity of traffic currently on show (based on the zoom level) and where this is relative to the entire bit-stream capture. Time information displayed is relative to the capture trigger time. Below this, the captured data and strobe signals are represented by different colours for each direction. The direction and signal type are indicated on the left side of the view. As the user zooms in and out, the detail and quantity of the signal information shown are adjusted. Users can insert two markers at points of interest by clicking the view. The timing of each marker is displayed along with the time between them.



Fig. 14 Bit-Stream View

The bit-stream view allows the user to debug problems at the signal level. During testing, other more high-level views may show errors e.g. the character view may display parity and disconnect errors. The bit-stream view can be used to investigate the data and strobe signals around the time at which the errors were detected. This may lead to the discovery of a signalling issue such as a simultaneous transition. The bitstream view screenshot below shows an example of a simultaneous transition.



Fig. 15 Simultaneous Transitions

For successful SpaceWire link initialisation, character synchronisation must occur. The bit-stream view can be used to visualise this as shown below.



Fig. 16 Character Synchronisation

Data and strobe lines of both ends are initially set low before one end (A) transmits Nulls (a specific sequence of eight bits) on which the other end (B) synchronises its receiver. Once synchronised, end B transmits Nulls on which end A synchronises its receiver. Further Nulls and FCTs are then exchanged according to the link initialisation Null/FCT handshake. The SpaceWire interfaces start at 10 Mbit/s before switching to the required operating speed (200 Mbit/s in this case).

XI. STATISTICS

The SpaceWire Link Analyser Mk3 software includes a status counter view that displays live statistics for traffic travelling in both directions on a SpaceWire link. This includes the signalling rate, the number of errors detected and a count of the different SpaceWire character types. Line charts graph these values as shown below.



Fig. 17 Status Counter View

The character/event count displayed can be set to the number detected per second or in total. The statistics can be paused to be examined or reset when required. If an error is detected, a red vertical line is shown at the appropriate time on the charts. The status counter display is useful for monitoring SpaceWire link utilisation and for detecting any errors during long tests. For example, it can be used to confirm no errors occurred. If any errors are detected, the same test can be rerun with a capture trigger set to occur when the error type identified is detected, allowing the problem to be investigated further.

XII. SAVE AND LOAD

SpaceWire traffic recorded by the SpaceWire Recorder is stored on hard disk and can therefore be opened and closed as required. In contrast the SpaceWire Link Analyser Mk3 uploads captured traffic to computer memory, meaning the user must choose to save this otherwise it will be discarded when the next capture takes place or the application is closed. Once saved this can be reopened. Being able to save and later open and view SpaceWire traffic allows the user to keep a record of SpaceWire traffic behaviour under different circumstances. It allows the user to save traffic containing unexpected errors and share this with others that can help with debug efforts.

XIII. CONCLUSION

SpaceWire traffic visibility is necessary to confirm SpaceWire equipment is operating correctly and to debug problems when these are encountered. The SpaceWire Link Analyser Mk3 and SpaceWire Recorder are two items of test equipment that provide comprehensive recording capabilities. The software provided with these displays the captured traffic in different levels of detail, providing valuable traffic visibility for debug and validation efforts. This paper described the SpaceWire Link Analyser Mk3 and SpaceWire Recorder hardware and the capabilities provided by these. The SpaceWire Link Analyser Mk3 captures detailed information about the bi-directional traffic travelling on a single SpaceWire link. It can capture up to 67 million events to on-board memory before it is quickly uploaded to a host PC using a USB 3.0 interface. The SpaceWire Recorder in comparison is a standalone machine, capable of recording significantly larger quantities of traffic on up to four SpaceWire links but in less detail. It records SpaceWire link data, time-codes and errors to a solid-state disk 1 TB in size as standard.

The software provided with the SpaceWire Link Analyser Mk3 and SpaceWire Recorder was also described, along with examples and descriptions of how these can help whilst testing SpaceWire equipment. The software for both units displays captured traffic in multiple common views in varying quantities and detail. The network view provides a high-level overview of large amounts of traffic, useful for identifying trends and for navigation. The packet view allows users to examine the packets transferred and can interpret and display the fields of RMAP packets. The character view is used to examine SpaceWire traffic at the character and exchange levels. The bit-stream view displays the data and strobe signals captured, useful for visualising data encoding.

The large quantities of traffic that can be captured by hardware, combined with new software capable of quickly navigating this and displaying it in multiple views with different detail, provides advanced SpaceWire traffic visibility that can aid those responsible for testing SpaceWire equipment.

References

- SpaceWire Remote Memory Access Protocol, ECSS-E-ST-50-52C, February 2010
- [2] STAR-Dundee, SpaceWire Link Analyser Mk3 User Manual, v1.00.
- [3] STAR-Dundee, SpaceWire Recorder User Manual, v1.01.
- [4] S. Parkes, https://www.stardundee.com/sites/default/files/SpaceWire%20User%27s%20Gui de.pdf, SpaceWire User's Guide, STAR-Dundee Website.
- [5] STAR-Dundee, https://www.stardundee.com/sites/default/files/SD_TN_004%20Link_Analyser_ Mk2_App_Note.pdf, SpaceWire Link Analyser Mk2: Debugging a SpaceWire Hardware Link Fault, STAR-Dundee Website
- [6] S. Mudie, C. McClements, A. Spark, S. Mills, A. Mason, M. Dunstan, S. Parkes, "Recording SpaceWire Traffic", SpaceWire Conference 2014.

Static Analysis for SpaceWire IP Cores

Test and Verification, Long Paper

Scott Calkins

Blue Pearl Software Shrewsbury, MA, USA scott.calkins@bluepearlsoftware

Abstract—SpaceWire IP core users often suffer common problems when using and modifying Space Wire IP cores. There is often little to no guidance as to what compiler messages are acceptable. Without running static analysis on these cores, designers run into development problems that hamper their ability to properly debug and enhance designs. Using the SpaceWire Link core that Goddard Space Flight Center maintains as a concrete example, a proven methodology will be detailed. In addition, this paper will explain the value of performing this analysis prior to enhancing that same SpaceWire core.

Index Terms—SpaceWire, IP, Lint, Clock Domain Crossing, CDC, Static Analysis, Symbolic Analysis, Structural Analysis, ASIC, FPGA, eASIC, Test, Simulation, Verification, Synthesis, Design Compiler, Blue Pearl Software, Waivers.

I. INTRODUCTION

There is a well-known problem in the IP space of SpaceWire IP: usability and pliability. Pliability can be a misleading term. We'll define this as the ability for the common engineer to rework the IP safely without injury to their system. Usability and pliability of IP cores in this space are hampering engineers from begin more efficient and effective. Cores can be made more effective by providing static analysis waivers that go along with their cores, and designers and pre-tag cores with static analysis messages as a baseline.

This industry has largely ignored a solution to this problem. Static analysis via structural analysis, as described in Scott Bloom's DVCon 2018 paper titled "Just Do it!- Who cares if a Structural Analysis tool is using Formal Verification"[1], is not necessarily running formal verification in order to get powerful results that are often missed using simulation alone. At that same conference it was noted that 65% of all testbench errors are human mistakes, not functional algorithmic mistakes. For instance, bugs introduced via cut and paste. Static analysis is a method for analyzing HDL designs by parsing and then synthesizing these designs into a technology independent structure where algorithms can be run to find common problems. Simulators and synthesizers parse the VHDL and Verilog code and translate it into a parse tree datastructure that their tools to use. As the DUT is being Scott Aron Bloom Blue Pearl Software Portland, OR, USA scott.bloom@bluepearlsoftware

parsed, or analyzed, both tools will report messages of varying severity to the user, these messages encompass the log file. The assumption is that hardware designers analyze the reports and log files and fix the issues. When a design runs without any messages, it is considered "clean." Yet, it is often the case, that the realities of release deadlines, in fact limit the ability of the designer to go over every issue. Instead time is allocated for verification analysis, and when that time is up, the designer moves on to other aspects of the design cycle.

II. RELEVANT BACKGROUND

Full disclosure regarding the authors. Scott Calkins serves as the World-Wide FAE Manager for Blue Pearl Software (BPS), while Scott Bloom is the Chief Technology Officer for BPS. Blue Pearl is a company that provides static analysis software for HDL designs.

In our experience, it is not uncommon, for real world designs, to have 10,000 to 50,000 messages when you combine the log files of Simulation and Synthesis. In a hypothetical scenario, you can work out why it is unrealistic to expect engineers to read through a logfile, let alone fix every issue. For the purposes of this paper, if an engineer spends 1-3 seconds on every message, analyzing 25,000 messages

25,000 messages * 2 sec of analysis time = 60,000 sec (1) 60,000 s = 16 + hours (2)

It is simply unreasonable to expect a designer to read through 16 hours of log files. More realistically, they will search for messages they have caused design problems in the past. This process must be repeated every time any portion of the design changes. Spending 2+ days per Simulation and Synthesis run for a typical design, is simply not an efficient use of project time or program assets.

III. SYMBOLIC SIMULATION

One form of static analysis is symbolic simulation. Back in the 1980s, there is a notion that Moore's Law would provide a problem for cycle-based simulation. In the 1985 paper, by R.E. Bryant, presented at the 1985 Chapel Hill Conference on VLSI. Bryant, delivered a paper often cited over the years [2]. Symbolic simulation takes years to come into the fold, mainly due to the fear over the large memory requirements to keep everything in state space. Yet the theory is solid, in 1990, R. E. Bryant writes again about the value of symbolic simulation.

"Symbolic simulation involves evaluating circuit behavior using special symbolic values to encode a range of circuit operating conditions. In one simulation run, a symbolic simulator can compute what would require many runs of a traditional simulator. Symbolic simulation has applications in both logic and timing verification, as well as sequential test generation. The concept of symbolic simulation has been discussed for over 10 years, but early attempts had only limited success. The recent introduction of more powerful, algorithmic methods of symbolic manipulation have had a major impact on the classes of circuits and properties that can be evaluated symbolically." [3]

Then, with the advent of cheaper hardware and memory for computing power, symbolic simulation verification was being used in many static analysis tools. However, symbolic simulation still needs a test pattern to run on the circuit. Often Automatic Test Pattern Generation (ATPG) is used in conjunction with Symbolic Simulation. Removing the need for the designer to drive the simulation engine, and allowing the EDA tool to focus on parts of the design where the simulator can be most effective.

As design size as continued to outpace the computing power required to minimize verification, performance and predictability are still questions that many have as recent as the mid 2000's. The value of symbolic simulation is that there is no user generated testbench, thus much of the time the user spends generating the framework and vectors to stimulate a design are removed, replaced instead with verification. While this does not solve issues that are functional in nature, getting their arms around the explosion of gates that must be simulated and the massive geometric computations that will have to be run on any given design starts to worry designers and tool vendors. [4]

Bryant stated in 2005. "By simulating a circuit symbolically, verification can avoid the combinatorial explosion that would normally occur when evaluating circuit operation over many combinations of input and initial state." [5]

IV. STRUCTURAL ANALYSIS

So, what is structural analysis? Structural analysis, as defined for this paper, is looking at the design in a clock independent method or at time zero without any change in the clock. Thus, time is static, where the primary focus is how the design is connected. When time is dynamic, or you want to see how your data changes from one clock cycle to another, you enter in type dynamic analysis, using a Symbolic or Cycle based simulator.

Structural analysis, also relieves the customer from driving the analysis via test vectors. An often-used analogy, structural analysis is "spell checking", whereas, dynamic is grammar checking.

Structural analysis will make sure all the words in my document are spelled correctly, however it requires a grammar

checking system to make sure that the combination of words make any sense and form constructive and meaningful sentences and paragraphs.

The mechanics of structural analysis are done in several ways, and at various stages of the design. By analyzing a design during and after the parsing stage of design analysis, many problems that can be found very quickly. After parsing, the second phase of analysis is done after the elaboration phase. This phase translates the HDL constructs into technology independent gates. As many structural issues require a more hardware related view of the design.

Let's looks at a case of analyzing Case Statements. Structural analysis can be used to find missing case items. A user driven method, using the brute force method, where a simulation testbench could be used to set the condition to every possible condition in the range, is an unrealistic approach to this problem. A simple 32 state One-Hot FSM shows the problem, no verification engineer is going to test every possible state. However, what about an encoded condition, where the blocks of bits are tied to different states of a communications channel. Verifying that you are not creating latches, for all signals being assigned, is still too expensive even for 128 different states when encoded across even 48 bits.

However, solving this using Structural Analysis, is a trivial problem using a Formal Verification engine. Using deterministic formal analysis that is static in nature, you can say:

assert(I = (!case1 && !case2 && !case3) (4)

Here, if any value is equal to case*x*, the formal engine solves it, proving that every possible value is covered by the case items. If a condition value is not covered, it can then be reported to the user. The advantage for the user, is simply turning on the check, as opposed to reading through the synthesis log for latch inferencing, and then spending considerable time to determine why the latch is created.

Another example is dangling net analysis. This can occur under a multitude of conditions. For instance, a signal that is defined and driven but is not read anywhere in the circuit. Structural Analysis is specifically designed to find these problems. Where the intent of the functionality is irrelevant, however the implementation is wrong independent of the algorithm. Back to our Spell Checker vs Grammar Checker analogy. An author using Kernel or Colonel needs a grammar checker to make sure if they are using the correct version. However, if they spelled it "Kolenel", they are wrong in any usage. Just as an FSM that relies on state 3 to transition to state 4 when a condition is triggered, may be algorithmically wrong. But if the next signal is not connected properly in state 3, it doesn't matter. Again, the user interaction is limited to turning on the dangling check. Not searching through simulation logs for X's

All forms of verification require the user to setup the EDA tool. Simulation requires a testbench, and the verification tool will only run on the portions of the design exercised by the testbench. Structural analysis does not rely on the designer to pick and choose what area of the design is being exercised,

instead the designer selects what analysis is to be performed on the design presented.

The benefit of this methodology is the ability to focus a subset of problems at any given time. In a similar fashion that a designer looks for "certain messages in a log file", they can focus on the issues that have caused the most problems. However, rather than looking for a needle in a haystack, the report will only contain needles.

Too often, verification engineers spend time working on problems that are not functional, not algorithmic, and not tied to the function of the design. They are the proverbial spelling errors. While these types of errors, might not present themselves as fatal errors. Using focused structural analysis, provides insight into the "hotspots" of problems in your design. The designer can then can quickly look at the VHDL or Verilog and determine the purpose of the block of code.

But what happens, when a designer determines that the issue found is not an issue? This can happen for number of reasons, it can be caused by the hardware designer doing something out of the ordinary, but safe with their a priori knowledge of the design. Or it could be caused by the the verification tool being too conservative and producing noise.

While verification tools are constantly improving their noise output, there are certain constructs the designer wants to simply ignore. Verification tools must provide a waivers system for this. A waiver is often considered a tag on an object with respect to a specific warning where the engineer is telling the tool that they want to allow this to exist, and not to report the issue any further.

Another example, is the design practice situation where the net has a load but no driver, this condition is often called "Unconnected Undriven and Used". This is often an X propagation error in your design, and typically a bug. Without fully inspecting the design and creating a testbench environment that specifically targets the loaded circuit, simulation based verification will not find this problem.

Case in point, a design from a Blue Pearl Customer, in which the functionality was considered complete, with full simulation verification being performed, had over 1200 signals where X's were being propagated, and yet no simulation reported the problem.

V. FORMAL VERIFICATION

Formal Verification (FV) is another form of verification, that goes beyond simulation. Formal Verification mathematically proves the Boolean logic you ask it to. It can report "yes its proven true", or "yes its proven false". However, it can also report "no, it can't be proven." The designer is giving up deterministic results, where you might not be able to find the results. For the very likely potential, that you may get no answer at all.

The biggest concern with Formal Verification tools, is not the likelihood of "unproven questions", is it still involves significant user interaction on the users part to get any answers at all.

The designer turns in simulation testbench generation, where they would inspect the code, and attempt to create

vectors to cover as much of the code as possible. That is changed out for inspecting the code and attempting to create assertions to cover as much of the code as possible.

Real world experience has shown, that while Formal Verification can find algorithmic intent issues that can't be found using simulation. It has not proven out to be any faster in user time. Often, it requires an expert in FV, in which the bar to become an expert is much higher than simulation.

An acceptable solution for the most common issues, is the rely on the Structural Analysis tool, to use FV as an engine to find issues. Much like it might choose to use Cycle Based simulator engine to determine the extent of a misconnect net.

Formal Verification tools are expensive. Not only from a fiscal budgetary view, but from a runtime as well as developer time budget as well. If you are fortunate to be working on a project that has infinite cash on hand for verification tools, infinite time to setup the tools, and infinite time to run the tools. Then by all means, run exhaustive simulation and formal verification runs. While there are ASIC designs with "infinite" budgets, one of the primary attractions of FPGAs has always been reduction in costs of the total design process. Unfortunately, this often means, budgets for verification are fare from infinite. Fortunately, Structural Analysis covers large portions of real world issues, found on FPGAs (and even ASICs).

VI. WAIVERS

The ability to waive issues found during analysis is critical to successfully verifying using structural analysis. To understand waivers, the static analysis system must be explained.

For static analysis, the design is parsed then synthesized. While analysis For every check that has been enabled by the designer, the analysis algorithm is performed at the appropriate level, producing a message for every enabled issue that is found. The parsing, translating and synthesis creates a database of objects, each object is tied to the code and to a generically mapped set of gates.

When a message is reported, the option is to waive that message or messages, most tools enable multiple messages to be waived via regular expression matching, the message can be flagged as a Waived message. This message is now tied to this object for this algorithm. The next time the algorithm is run, and the message is to be reported, it will be checked against the waiver system to see if the message should be reported to the user.

Waivers do not disable the analysis, and they are stored in the persistent model of the message system. This enables a single waiver waive messages over multiple runs of their design. It also enables the opposite, for a user to remove a waiver from the system, viewing the message in previous runs.

When reviewing a 25,000-line synthesis report, the signal to noise ratio, is simply too low. You spend too long reading trying to find valid messages. Structural analysis is often criticized for having smaller, but also noisy reports. Using waivers enables the designer to increase the SNR by reducing the noise. Unfortunately, simulation and synthesis simply do not provide the mechanism to waive messages.

A waiver system allows you to spend the 36 hours going over these messages once and for all. Then, when the code has changed, there is no question as to what is new and what has been analyzed already.

VII. SPACEWIRE LINK RESULTS

The Goddard Space and Flight Center's SpaceWire cores are being used at Orbital – ATK. When working with ATK to examine the full IP of the design, it was decided to focus on the Link, where it was determined the majority of the designs Clock Domain Crossings (CDCs) existed.

In designs with multiple clocks, its critical to make sure that data does not cross from one clock domain to another without proper synchronization. However, its extremely difficult to find CDCs via simulation or using Formal Verification tools. To do so, you must introduce for every crossing, jitter on the clocks, and then verify the data path has not introduced errors due to the jitter.

In our previous analysis engagements with Orbital-ATK, we already knew where the CDCs in the design were primarily located.

The Link core has been in use as a core in multiple designs. However, under analysis Blue Pearl found 31 CDC errors, in one design and 14 in another. A CDC error is defined as a Clock Domain Crossing, where the data is not being properly synchronized at the crossing. These were not CDCs inside the IP block, but rather near the instantiation of the Link IP block.

The investigation leads us to run a full CDC analysis on the core itself. The results, were as follows:

- 48 CDCs
 - o 47 Synchronized crossings
 - 1 Unsynchronized crossing
- 53 signals driven and not used
 - \circ 20 are nets
 - o 33 are pins
- 507 assignments previously assigned
- 9 signals that have no reset at all
- 380 asynchronous resets that have no synchronous deassertion

The single unsynchronized CDC in the Link core was the issue. Upon further inspection, this signal is static. That is, it's a signal capable of being dynamic, but is functionally static. A signal that the processor sets or a set of external pins determines. This is a CDC that cannot cause metastability issues, therefore applying a waiver to this CDC is warranted. By adding waiver on this CDC, the next designer will not need to investigate this. It will no longer be reported as a problem. Now the signals that were found in error (warnings to be specific) had no baseline. When CDC is run on the top level design, CDCs that are caused by the core issue, will no longer be reported. Reducing the noise significantly. Any CDC reported using the core, need to be investigated for synchronization, and correct.

The Link also had significant structural issues as well? 53 signals that are not used but are driven. Many were tied to IP,

where an IP core drives a signal, but the designer is not using that portion of the IP. However, some of the dangling but driven signals, should have been connected, or were valid signals but unneeded for the specific design. Waivers are also appropriate for these types of issues. But a waiver on each of these explaining why they are not connected will greatly enhance the reusability of this core. Creating a proper set of waivers for your IP cores, not just purchased cores, but cores being reused by your own teams, enables better quality verification on the next usage of the IP.

The analysis of the Resets issues, it was found that the intent of signals was to not be resettable. It was made clear via documentation. This message could have been avoided using naming techniques or configuring the structural analysis to ignore them. It is not uncommon for FPGA designs to have datapath signals, where the fabric requires these signals to be without reset. The FPGA replaces the value immediately, so a reset is unnecessary. This condition, begs for a waiver against that message on those signals.

Modern waivers systems not only track the conditions of the waiver in a reproducible manner. They also track the reason why the waiver when done correctly have a reason, a message, stating why it is to be waived and not analyzed further. This documents the waiver and can be reported later and analyzed during a design review. Understanding why an engineer decided to waive a message gives further insight into the design itself.

The remaining reset issues, were a mix of various issues in Microsemi cores, also to be waived.

When "Previously assigned signals" are reported, often it points to poor code, that can either create latches or glitches. Having 507 of them, is a red flag for a design issue. It means the signal in question is assigned twice. Often in a statement after the reset, outside of any if statement, then later in the process block, assigned within an if statement or a case statement with some specific logical condition to change the value.

This is a very risky and leads to conditions that "sometimes work", meaning it may work at 200Ghz, but not 400. Why risk having these issues in your design? Experience tells us, fix them all. These issues were not done by anyone from Goddard, so it was impossible to determine the reasoning for the code. Without a waiver, every time the code is reused, the issues will be suspect. Using waivers to document your reasons for nonconforming further documents your design, enabling easier reuse in further designs.

The final check, that warrants discussion, are the 380 missing deassertion of reset. From what could be determined by inspection, deassertion circuit is required to be outside the Link core. Waiving these issues, will remove the noise from the IP block when its run in another design.

VIII. CONCLUSION

We hope that the takeaway of this paper is to convince everyone in the SpaceWire community to analyze their designs with static analysis and place sensible waivers on all messages to aid the reuse of these cores. One of the major values of a value of a properly waived IP block is the lower cost of reuse. It enables designers to focus on real issues, that are new, and not issues that have previously existed and have been properly diagnosed. Maintaining a clear analysis result, will produce less messages during simulation and synthesis. The reduction in engineer time will be significant.

Furthermore, static analysis is the only place to find CDC issues, and these are the stuff of nightmares for engineers. Utilizing a static analysis tool will save time and make for a more efficient design flow on any design, let alone cores and IP designed for space, where we all know failures can result in massive losses.

ACKNOWLEDGMENT

The author wants to acknowledge the following individuals for their time and assistance with the investigation of the SpaceWire cores: Marco Figueiredo from Orbital – ATK, Doug Carssow from the Naval Research Labs, and Michael Pagen from NASA, Goddard Space Flight Center.

REFERENCES

- [1] A. Bloom, Scott, "Just Do it!- Who cares if a structural snalysis tool is using formal verification" DVCon, February 2018.
- [2] E. Bryant, Randal, "Symbolic verification of MOS circuits," 1985 Chapel Hill Conference on VLSI, 1985, pp. 419–438.
- [3] E. Bryant, Randal, "Symbolic Simulation—Techniques and Applications" http://repository.cmu.edu/compsci/240/, 1990
- [4] Bertacco, Valeria, "Scalable hardware verification with symbolic simulation", Springer, 2006.
- [5] E. Bryant, Randal "Verification of synchronous circuits by symbolic simulation" Conference paper, Hardware Specification, Verification and Synthesis: Mathematical Aspects pp 14-24. Part of the Lecture Notes in Computer Science book series (LNCS, volume 408), First Online: 31 May 2005.

Author Surname A - K

T. Bahls, M. Bihler, S. Tarassenko; SPACEWIRE MEETS BIG DATA – REALTIME DATA MINING	139
T.Bahls, M. Bihler; USING SPACEWIRE TIME-CODES FOR GLOBAL SYNCHRONIZATION OF PLL-BASED LOCAL CLOCKS	109
G. Baterina, A.Senior; GALVANIC ISOLATION OF SPACEWIRE PORTS – AN INNOVATIVE DESIGN APPROACH	84
M. Bihler, T. Bahls; EFFICIENT IMPLEMENTATION OF ON-CHIP COMMUNICATION OPTIMIZED FOR SPACEWIRE NETWORKS	113
S. Calkins, S. Bloom; STATIC ANALYSIS FOR SPACEWIRE IP CORES	247
S. Clancy, J. Lux; ABSTRACTED SPACEWIRE INTERFACE FOR DEEP SPACE RADIO	107
S. Clancy, M. Chase, A. Yarlagadda, M. Starch, J. Lux; SPACEWIRE AS A CUBE-SAT INSTRUMENT INTERFACE	26
C. P. Collier, B. Ripley; SPACEVNX: A SCALABLE AND HIGHLY FLEXIBLE SMALL FORM FACTOR STANDARD FOR SMALL-SATS AND CUBESATS	167
C. P. Collier; A SWAPC SCALABLE HIGH PERFORMANCE HARDWARE STANDARD FOR 6U/3U VPX BASED PROCESSING AND I/O SYSTEMS	210
D. DeLazari, A. Deucher, A. Santos, S. Finco, A. Horn, M. Beer, V. Ohlen; DEBUG AND VERIFICATION OF SPACEWIRE LINKSS. Eguchi, T. Takashima, I. Shinohara; DEVELOPMENT OF THE NEW MISSION DATA RECORDER (MDR)	17
WITH SPACEWIRE INTERFACE ONBOARD THE ERG SPACECRAFT AND THE RESULT OF ITS PERFORMANCE ON ORBITS	159
S. Eguchi, T. Takashima, I. Shinohara; SPACEWIRE NETWORK SYSTEM FOR ERG MISSION NETWORK AND THE RESULT OF ITS PERFORMANCE	123
A. Ferrer-Florit, A. Gonzalez Villafranca, S.Parkes, C. McClements; SPACEFIBRE INTERFACE AND ROUTING SWITCH IP CORES	50
R. Ginosar, P. Aviely, R. Nesher, Z. Meister, T. Israeli, D. Reznik; TESTING AND VALIDATION OF SPACEWIRE CONTROL AND DATA LINKS OF RC64	228
D. González, J. López, A. Frouin, J. Beister, D. Levacq; SPACE-QUALIFIED EUROPEAN LVDS COMPONENTS FOR SPACEWIRE NETWORKS	66
S. Hermant, K. Enouf; THE EVOLUTION OF SPACEWIRE ELECTRICAL INTERCONNECT	37
H. Hihara, M. Kuribayashi, K. Murozono, K. Yamada, Y. Otake, K. Kobayashi, Y. Tsukita, H. Marouka, J. Furuta; PROGRAMMABLE SPACEFIBRE INTERFACE WITH NANOBRIDGE FIELD PROGRAMMABLE GATE ARRAY	92
R. Johannes; COMPARATIVE PERFORMANCE OF VITA 78 CONNECTOR SYSTEMS FOR DAUGHTER CARD TO BACKPLANE APPLICATION	70
R. Johannes; MODULAR INTERCONNECT FOR POINT TO POINT AND BACKPLANE SPACE	74
Y. Junhui, N. Yuehua, L. Xiaojuan; COMMUNICATION PROTOCOL BASED ON TIME MULTIPLEXING FOR SPACEWIRE DATA HANDLING NETWORKS	163
M. Karppinen, A. Tanskanen, J. Ollila, D. Lopez Molina, J. Barbero, C. Boatella Polo, R. Jansen, I. McKenzie; RADIATION TOLERANT OPTICAL TRANSCEIVERS FOR SPACEFIBRE DATA LINK	87
Author Surname L - Z

F. Lange, R. Ginosar, P. Aviely, T. Israeli; IMAGING NOGAH SOFTWARE-DEFINED SYSTEM COMPRISING MANY RC64 PROCESSORS FOR OPTICAL AND SAR OBSERVATION SATELLITES	3 22
F. Lange, R. Ginosar, T. Israeli, G. Danin, M. Goren, P. Aviely; WIDEBAND DVB-S2X FOR EOS DOWNLINK ON SPACEFIBRE-INTERCONNECTED DUAL RC64	148
I. Lavrovskaya, I. Korobkov; APPROACH TO INCREASE SPACEFIBRE LINK BANDWIDTH USAGE FOR STREAMING DATA TRANSFER I. Lavrovskaya, Y. Sheynin, V. Olenev, I. Korobkov, L. Kurbanov, D. Dymov; COMPUTER-AIDED DESIGN	143
I. Lavrovskaya, E. Suvorova, A. Khakhulin, I. Orlovsky, Y. Sheynin, I. Korobkov, V. Olenev; REAL TIME VIDEO DATA TRANSMISSION IN SPACEFIBRE NETWORKS WITH THE ESDP TRANSPORT PROTOCOL	220
A. Leoni, L. Fanucci, D. Jameux; SIMULATOR FOR HIGH-SPEED NETWORKS (SHINE): AN OMNET++ SIMULATOR FOR SPACEFIBRE AND SPACEWIRE NETWORKS R. T. Logan; PHOTONIC TRANSCEIVERS FOR SPACEFIBRE DATALINKS	128 56
J. Marshall; STANDARDIZED HIGH PERFORMANCE SPACEVPX MODULES LEVERAGE SPACEWIRE AS AN INTERNAL/EXTERNAL CONTROL FABRIC	102
 S. Mills, C. McClements, D. Paterson, P. Scott, S. Parkes; TESTING OVER ETHERNET WITH THE SPACEWIRE GBE BRICK S. Mudie, D. Gibson, C. McClements, S. Mills, S. Parkes; SPACEWIRE LINK ANALYSER MK3 AND SPACEWIRE RECORDER 	11 240
S. Mudie, D. Gibson, C. McClements, S. Mills, S. Parkes; TESTING SPACEFIBRE EQUIPMENT AND SYSTEMS	233
S. Parkes, A. Srivastava, C. McClements, P. Scott, D. Dillon, A. Ferrer Florit, A. Gonzalez Villafranca; SPACEFIBRE CAMERA	187
S. Parkes, P. Scott, D. Dillon, M. Dunstan, C. McClements, A. Ferrer Florit, A. Gonzalez Villafranca; SPACEVPX-RTG4 BOARD WITH SPACEWIRE OR SPACEFIBRE BACKPLANE	193
D. Poudereux, J. Barbero, J. M. G. Tijero, I. Esquivias, I. Mackencie; OPTICAL SWITCHES WITH NO MOVING PARTS FOR SPACE APPLICATIONS	173
K. Romanowski, W. Mich, P. Tyczka, R. Renk, V. Kollias, N. Pogkas; EVALUATION AND INTEROPERABILITY TESTING OF THE SPACEWIRE-R PROTOCOL	211
M. Ruiz, J. B. Feron; CONTROL LOOP PROCESSOR: A RELIABLE AND AGILE PROCESSING PLATFORM FOR MISSION CRITICAL APPLICATIONS	78
T. Sasaki, S. Hirakuri; SPACEWIRE AND SPACEFIBRE FOR INTERNAL ARCHITECTURE IN NVDRs	98
H. J. Sedlmayr, R. Bayer, A. Beyer, M. Maier, N. Seitz, M. Chalon, W. Bertleff, W. Friedl, T. Obermeier; SPACEHAND: A MULTI-FINGERED ROBOTIC HAND FOR THE USE IN SPACE	31
F. Siegle, A. Leoni; STANDARDISATION EFFORTS FOR A NETWORK MANAGEMENT AND DISCOVERY PROTOCOL FOR SPACEFIBRE	133
T. Solokhina, J. Petrichkovich, A. Glushkov, L. Menshenin, D. Kuznetsov, S. Parkes, D. Dymov; RADIATION TOLERANT MICROPROCESSOR FOR THE COMPUTER VISION WITH SPACEFIBRE LINKS	168
D. Thurnes; SPACEWIRE AND SPACEFIBRE OVERVIEW AND ROADMAP	209
M. Tomitaka, Y. Igarashi, S. Ichikawa, N. Inaba, A. Tomiki, K. Matsuzaki, R. Kobayashi, M. Kumakiri, I. Fujishiro, M.Nomachi; FEASIBILITY STUDY OF WIRELESS COMMUNICATION SYSTEM OPERATING ON SPACEWIRE NETWORK	118

J. J. Wang, N. Rezzak, S. Varela, K. O'Neill, A. Gu, E. Hamdy; SERDES SINGLE EVENT EFFECTS IN 65NM	
FLASH-BASED RTG4 FPGA	177
E. Weih, R. Wiest, O. Ried, P. Rastetter, M. Stahle, G. Lohse, M. Steiner; SPACEFIBRE BASED MASS MEMORY – EXTREME RAPID MASS MEMORY UNIT (EXTRA MMU) ENABLED BY SPACEFIBRE	180
N. J. Wessman, F. Johansson, F. Hernandez, J. Andersson, C. Monteleone, R. Weigand; STATUS UPDATE ON NEW STANDARD PRODUCTS WITH SPACEWIRE SUPPORT	43
Y. Xiaosu, Z. Huasong, Z. Chunxi; IMPLEMENTATION OF SPACEWIRE OPTICAL FIBER LINK FOR HIGH-SPEED AND LONG-DISTANCE DATA ACOULSITION	154
	10.

Tuesday 15th May

Test & Verification (Short Papers)	
S. Mills, C. McClements, D. Paterson, P. Scott, S. Parkes; TESTING OVER ETHERNET WITH THE SPACEWIRE GBE BRICK	11
D. DeLazari, A. Deucher, A. Santos, S. Finco, A. Horn, M. Beer, V. Ohlen; DEBUG AND VERIFICATION OF SPACEWIRE LINKS	17
Missions & Applications (Long & Short Papers)	
F. Lange, R. Ginosar, P. Aviely, T. Israeli; IMAGING NOGAH SOFTWARE-DEFINED SYSTEM COMPRISING MANY RC64 PROCESSORS FOR OPTICAL AND SAR OBSERVATION SATELLITES	22
S. Clancy, M. Chase, A. Yarlagadda, M. Starch, J. Lux; SPACEWIRE AS A CUBE-SAT INSTRUMENT INTERFACE	26
H. J. Sedlmayr, R. Bayer, A. Beyer, M. Maier, N. Seitz, M. Chalon, W. Bertleff, W. Friedl, T. Obermeier; SPACEHAND: A MULTI-FINGERED ROBOTIC HAND FOR THE USE IN SPACE	31
Components 1 (Long Papers)	
S. Hermant, K. Enouf; THE EVOLUTION OF SPACEWIRE ELECTRICAL INTERCONNECT	37
N. J. Wessman, F. Johansson, F. Hernandez, J. Andersson, C. Monteleone, R.Weigand; STATUS UPDATE ON NEW STANDARD PRODUCTS WITH SPACEWIRE SUPPORT	43
Components 2 (Long Papers)	
A. Ferrer-Florit, A. Gonzalez Villafranca, S. Parkes, C. McClements; SPACEFIBRE INTERFACE AND ROUTING SWITCH IP CORES	50
R. T. Logan; PHOTONIC TRANSCEIVERS FOR SPACEFIBRE DATALINKS	56

Wednesday 16th May

Components (Short Papers)

D. González, J. López, A. Frouin, J. Beister, D. Levacq; SPACE-QUALIFIED EUROPEAN LVDS COMPONENTS FOR SPACEWIRE NETWORKS	66
R. Johannes; COMPARATIVE PERFORMANCE OF VITA 78 CONNECTOR SYSTEMS FOR DAUGHTER CARD TO BACKPLANE APPLICATION	70
R. Johannes; MODULAR INTERCONNECT FOR POINT TO POINT AND BACKPLANE SPACE	74
M. Ruiz, J. B. Feron; CONTROL LOOP PROCESSOR: A RELIABLE AND AGILE PROCESSING PLATFORM FOR MISSION CRITICAL APPLICATIONS	78
G. Baterina, A. Senior; GALVANIC ISOLATION OF SPACEWIRE PORTS – AN INNOVATIVE DESIGN APPROACH	84
M. Karppinen, A. Tanskanen, J. Ollila, D. Lopez Molina, J. Barbero, C. Boatella Polo, R. Jansen, I. McKenzie; RADIATION TOLERANT OPTICAL TRANSCEIVERS FOR SPACEFIBRE DATA LINK	87
H. Hihara, M. Kuribayashi, K. Murozono, K. Yamada, Y. Otake, K. Kobayashi, Y. Tsukita, H. Marouka, J. Furuta; PROGRAMMABLE SPACEFIBRE INTERFACE WITH NANOBRIDGE FIELD PROGRAMMABLE	02
	94

On-board Equipment & Software (Short Papers)

T. Sasaki, S. Hirakuri; SPACEWIRE AND SPACEFIBRE FOR INTERNAL ARCHITECTURE IN NVDRs	98
J. Marshall; STANDARDIZED HIGH PERFORMANCE SPACEVPX MODULES LEVERAGE	
SPACEWIRE AS AN INTERNAL/EXTERNAL CONTROL FABRIC	102
S. Clancy, J. Lux; ABSTRACTED SPACEWIRE INTERFACE FOR DEEP SPACE RADIO	107

Networks & Protocols (Short Paper)

T. Bahls, M. Bihler; USING SPACEWIRE TIME-CODES FOR GLOBAL SYNCHRONIZATION OF PLL-BASED LOCAL CLOCKS	109
M. Bihler, T. Bahls; EFFICIENT IMPLEMENTATION OF ON-CHIP COMMUNICATION OPTIMIZED FOR SPACEWIRE NETWORKS	113
M. Tomitaka, Y. Igarashi, S. Ichikawa, N. Inaba, A. Tomiki, K. Matsuzaki, R. Kobayashi, M. Kumakiri, I. Fujishiro, M. Nomachi; FEASIBILITY STUDY OF WIRELESS COMMUNICATION SYSTEM OPERATING ON SPACEWIRE NETWORK	118
S. Eguchi, T. Takashima, I. Shinohara; SPACEWIRE NETWORK SYSTEM FOR ERG MISSION NETWORK AND THE RESULT OF ITS PERFORMANCE	123
A. Leoni, L. Fanucci, D. Jameux; SIMULATOR FOR HIGH-SPEED NETWORKS (SHINE): AN OMNET++ SIMULATOR FOR SPACEFIBRE AND SPACEWIRE NETWORKS	128
F. Siegle, A. Leoni; STANDARDISATION EFFORTS FOR A NETWORK MANAGEMENT AND DISCOVERY PROTOCOL FOR SPACEFIBRE	133
Poster Presentations	
T. Bahls, M. Bihler, S. Tarassenko; SPACEWIRE MEETS BIG DATA – REALTIME DATA MINING	139
I. Lavrovskaya, I. Korobkov; APPROACH TO INCREASE SPACEFIBRE LINK BANDWIDTH USAGE FOR STREAMING DATA TRANSFER	143
F. Lange, R. Ginosar, T. Israeli, G. Danin, M. Goren, P. Aviely; WIDEBAND DVB-S2X FOR EOS DOWNLINK ON SPACEFIBRE-INTERCONNECTED DUAL RC64	148
Y. Xiaosu, Z. Huasong, Z. Chunxi; IMPLEMENTATION OF SPACEWIRE OPTICAL FIBER LINK FOR HIGH-SPEED AND LONG-DISTANCE DATA ACQUISITION	154
S. Eguchi, T. Takashima, I. Shinohara; DEVELOPMENT OF THE NEW MISSION DATA RECORDER (MDR) WITH SPACEWIRE INTERFACE ONBOARD THE ERG SPACECRAFT AND THE RESULT OF ITS	
PERFORMANCE ON ORBITS	159

Y. Junhui, N. Yuehua, L. Xiaojuan; COMMUNICATION PROTOCOL BASED ON TIME MULTIPLEXING FOR SPACEWIRE DATA HANDLING NETWORKS 163

C. P. Collier, B. Ripley; SPACEVNX: A SCALABLE AND HIGHLY FLEXIBLE SMALL FORM FACTOR STANDARD FOR SMALL-SATS AND CUBESATS

T. Solokhina, J. Petrichkovich, A. Glushkov, L. Menshenin, D. Kuznetsov, S. Parkes, D. Dymov; RADIATION TOLERANT MICROPROCESSOR FOR THE COMPUTER VISION WITH SPACEFIBRE LINKS 168

167

D. Poudereux, J. Barbero, J. M. G. Tijero, I. Esquivias, I. Mackencie; OPTICAL SWITCHES WITH NO MOVING PARTS FOR SPACE APPLICATIONS 173

J. J. Wang, N. Rezzak, S. Varela, K. O'Neill, A. Gu, E. Hamdy; SERDES SINGLE EVENT EFFECTS IN 65NM FLASH-BASED RTG4 FPGA 177

Thursday 17th May

On-board Equipment & Software (Long Papers)

E. Weih, R. Wiest, O. Ried, P. Rastetter, M. Stahle, G. Lohse, M. Steiner; SPACEFIBRE BASED MASS MEMORY – EXTREME RAPID MASS MEMORY UNIT (EXTRA MMU) ENABLED BY SPACEFIBRE	180
S. Parkes, A. Srivastava, C. McClements, P. Scott, D. Dillon, A. Ferrer Florit, A.Gonzalez Villafranca; SPACEFIBRE CAMERA	187
S. Parkes, P. Scott, D. Dillon, M. Dunstan, C. McClements, A. Ferrer Florit, A. Gonzalez Villafranca; SPACEVPX-RTG4 BOARD WITH SPACEWIRE OR SPACEFIBRE BACKPLANE	193
Networks & Protocols (Long Papers)	
I. Lavrovskaya, E. Suvorova, A. Khakhulin, I. Orlovsky, Y. Sheynin, I. Korobkov, V. Olenev; REAL TIME VIDEO DATA TRANSMISSION IN SPACEFIBRE NETWORKS WITH THE ESDP TRANSPORT PROTOCOL	200
D. Thurnes; SPACEWIRE AND SPACEFIBRE OVERVIEW AND ROADMAP	209
C. P. Collier; A SWAPC SCALABLE HIGH PERFORMANCE HARDWARE STANDARD FOR 6U/3U VPX BASED PROCESSING AND I/O SYSTEMS	210
K. Romanowski, W. Mich, P. Tyczka, R. Renk, V. Kollias, N. Pogkas; EVALUATION AND INTEROPERABILITY TESTING OF THE SPACEWIRE-R PROTOCOL	211
Test & Verification 1 (Long Papers)	
I. Lavrovskaya, Y. Sheynin, V. Olenev, I. Korobkov, L. Kurbanov, D. Dymov; COMPUTER-AIDED DESIGN SYSTEM FOR ONBOARD SPACEWIRE NETWORKS	220
R.Ginosar, P. Aviely, R. Nesher, Z. Meister, T. Israeli, D. Reznik; TESTING AND VALIDATION OF SPACEWIRE CONTROL AND DATA LINKS OF RC64	228
Test & Verification 2 (Long Papers)	
S. Mudie, D. Gibson, C. McClements, S. Mills, S. Parkes; TESTING SPACEFIBRE EQUIPMENT AND SYSTEMS	233
S. Mudie, D. Gibson, C. McClements, S. Mills, S. Parkes; SPACEWIRE LINK ANALYSER MK3 AND SPACEWIRE RECORDER	240
S. Calkins, S. Bloom; STATIC ANALYSIS FOR SPACEWIRE IP CORES	247



4LINKS Limited

4Links designs, manufactures, supplies and supports an extensive range of SpaceWire test equipment, network simulators, development systems and IP products.

The 4Links product range is renowned as being the most comprehensive and reliable on the market. Our Modular Satellite Development Systems enables very early integration. Thus avoiding the many problems that occur when integration is only carried out long after all the design decisions for each subsystem have been made, when it is too late to fix incompatibilities.

Our SpaceWire routing switch and codec IP products, using a patented synchronous receiver design, are implemented in a single clock domain and thus avoid all of the problems of asynchronous logic and the requirement for careful FPGA placement. This IP could be critical to the success of your next SpaceWire project!

4Links is based in the Science and Innovation Centre at Bletchley Park, the World War II codebreaking centre, in Buckinghamshire, UK.



COBHAM SEMICONDUCTOR SOLUTIONS

Cobham Semiconductor Solutions provides HiRel standard products, ASICs, and radiation testing services. Our Cobham Gaisler site in Goteborg, Sweden provides IP cores and supporting development tools for embedded processors based on the SPARC architecture along with SpaceWire Routers and boards.

The key product is the LEON synthesizable processor model together with a full development environment and a library of IP cores (GRLIB). Our personnel have extended design experience, and have been involved in establishing European standards for ASIC and FPGA development. Cobham Gaisler has extensive experience in the management of ASIC development projects, and in the design of flight quality microelectronic devices. The company specializes in digital hardware design (ASIC/FPGA) for both commercial and aerospace applications.



AXON' CABLE

The Axon' group designs and manufactures wire, cable, connectors and cable assemblies for advanced technology applications in the principal fields of space, aeronautics, medical electronics, automotive and scientific research. Headquartered in France (100 Km east of Paris) the Group employs some 2000 staff in 14 subsidiaries across Europe, America and Asia, with an annual turnover of €140 million euro.

Axon' Cable has been involved in many space projects, including the International Space Station, many telecoms, scientific and earth observation satellites and rocket launchers including Vega and Ariane 5. The company enjoys space flight heritage dating back to 1997 and is currently completing all the cabling for the ExoMars Rover vehicle. Axon' has also been selected as the main interconnect partner for the OneWeb satellite mega constellation.

The group offers various types of products for space applications:

- ESCC approved wires, cables and connectors,
- lightweight aluminium round cables and braids,
- aluminium bus bars for satellite power distribution,
- MIL-STD-1553B databus looms for digital transmission systems,
- high data rate links for Voice-Data-Image transmission including SpaceWire, SpaceFibre, IEEE1394, EtherSpace, WizardLink and Fibre Channel,
- and custom-designed products for specific applications.

Additionally, Axon' has been selected either as prime or subcontractor on a number of ESA and CNES technology research projects including:

- High temperature thruster cables,
- Low mass SpaceWire,
- EMC shielding techniques
- Nano-D connectors
- Miniature Power connectors
- SpaceFibre connectors
- New (MicroMach) SpaceWire connector
- ESD anti-static cables

Glenair.

GLENAIR

Glenair - out of this world of interconnect solutions

SpaceWire cable assemblies:

Glenair offer a complete range of SpaceWire cable assemblies for laboratory and flight use. In support of the SpaceWire protocol Glenair also offer a complete range of Micro D connectors for vacuum chamber and router interface use. For more information on Glenair's space products portfolio please contact: Ross Thomson, Business Manager - Space Interconnect Systems Glenair UK Ltd 40 Lower Oakham Way Oakham Business Park Mansfield, Nottinghamshire NG18 5BY, UK e-mail: <u>rthomson@glenair.co.uk</u> Office: +44 (0) 1623 638100 Mobile/ cell: +44 (0) 7711 029 715

SpaceFibre:

Glenair designs and manufactures a full range of fiber-optic interconnect products to support spacecraft systems.

These include radiation-tolerant high-speed opto-electronic transceivers supporting SpaceFibre, sRIO and other high-speed protocols up to 10 Gbps per lane, as well as fiber-optic cable assemblies, connectors, inspection and cleaning kits, and training of personnel to insure mission success. For more information on Glenair's fibre optic product portfolio and capability please contact: Ronald T. Logan Jr., Ph.D.

Chief Technologist, Sr. Director Active Components Glenair Inc. 1211 Airway,Glendale California 91202 -2497, USA e-mail: <u>rlogan@glenair.com</u> Office: +1 818 247 6000

www.glenair.com



MICROSEMI CORPORATION

Microsemi Corporation (Nasdaq: MSCC) offers a comprehensive portfolio of semiconductor and system solutions for aerospace & defense, communications, data center and industrial markets. Products include high-performance and radiation-hardened analog mixed-signal integrated circuits, FPGAs, SoCs and ASICs; power management products; timing and synchronization devices and precise time solutions, setting the world's standard for time; voice processing devices; RF solutions; discrete components; enterprise storage and communication solutions, security technologies and scalable anti-tamper products; Ethernet solutions; Power-over-Ethernet ICs and midspans; as well as custom design capabilities and services. Microsemi is headquartered in Aliso Viejo, Calif., and has approximately 4,800 employees globally.



BLUE PEARL SOFTWARE

Blue Pearl Software is a provider of design automation software for ASIC, FPGA and IP RTL verification. Our customers are RTL managers and developers in military, aerospace as well as other safety critical markets who wish to avoid costly and time-consuming design spins due to simulation versus hardware mismatches, invalid timing constraints and clocking issues that can cause metastability issues in hardware. The companies Visual Verification Suite is customer proven on SpaceWire designs, finding issues during RTL development and IP integration that would have resulted in design spins or worse yet design failure in the field.

The Visual Verification Suite complements RTL simulation solutions with an integrated easy-to-use, static and formal linting solution and provides advanced debug, clock and clock domain crossing analysis and automated SDC generation. The suite's Management Dashboard generates progress reports for audits and design reviews as well as ensuring that all tests have been completed and passed prior to tape out and final signoff. The Suite has been architected to deliver the fastest bug find / fix rate in the industry improving productivity, reducing risk while decreasing development costs.

Contact Info

- General Information: info@bluepearlsoftware.com
- Customer Support: support@bluepearlsoftware.com
- Website: <u>www.bluepearlsoftware.com</u>



STAR-DUNDEE Limited

STAR-Dundee is an aerospace engineering company, which designs network and related data-handling technology for use on-board spacecraft. STAR-Dundee provides electronic test and development equipment and chip designs for spaceflight applications.

Our highly experienced engineers were instrumental in the development of SpaceWire, writing the ECSS standard with inputs from international spacecraft engineers. SpaceWire is now widely used on-board spacecraft with over 100 space missions already in orbit or currently being designed using SpaceWire technology. Our engineers are currently leading the research, technical development and standardisation of the next generation of SpaceWire technology, SpaceFibre, which is a substantial leap forward, offering much higher data rates, quality of service, fault detection, isolation and recovery, deterministic data delivery, low latency time-synchronisation and event signalling, and many other features and benefits.

Since 2002, STAR-Dundee has provided SpaceWire evaluation, test and development equipment to the world's space agencies and aerospace companies. Our SpaceWire interface boards and units are used in Electronic Ground Support Equipment (EGSE) for integrating and testing many spacecraft. Our IP cores are integrated in spaceflight systems monitoring the Earth, exploring our Solar System, studying the universe and supporting commercial space applications.

STAR-Dundee is committed to providing the best possible solution for your application. Our team of highly qualified and experienced engineers understands the challenges of designing systems for space applications. Our well proven technology has flown on many high profile space missions. Part of our commitment to our customers is the effort that we spend on the research, development and standardisation of data-handling technology. SpaceFibre is the latest manifestation of our commitment to engineering excellence and international standardisation.













BACKGROUND IMAGE: ESA/HUBBLE & NASA; CC BY 4.0 WWW.ESA.INT/SPACEINIMAGES/IN AGES/2017/06/A_STORMY_STELLAR_NU AVID SERRANO: CC BY-SA 3.0 IONS.WIKIMEDIA.ORG/WIKI/FILE:L HTTPS://COM