SpaceFibre Evolution: 100 Gbps with Low-Overhead SoC Integration

Alberto Gonzalez Villafranca
STAR-Barcelona S.L.,
Av. Cerdanyola, 79-81, p.1 of.11
08172 Sant Cugat del Vallès
Barcelona, Spain
alberto.gonzalez@star-dundee.com

Albert Ferrer Florit
STAR-Barcelona S.L.,
Av. Cerdanyola, 79-81, p.1 of.11
08172 Sant Cugat del Vallès
Barcelona, Spain
albert.ferrer@star-dundee.com

Marti Farras Casas
STAR-Barcelona S.L.,
Av. Cerdanyola, 79-81, p.1 of.11
08172 Sant Cugat del Vallès
Barcelona, Spain
marti.farras@star-dundee.com

Dave Gibson
STAR-Dundee Ltd.,
STAR House, 166 Nethergate
Dundee, DD1 4EE, UK
dave.gibson@star-dundee.com

Andrew MacLennan
STAR-Dundee Ltd.,
STAR House, 166 Nethergate
Dundee, DD1 4EE, UK
andrew.maclennan@star-dundee.com

Stuart Mills
STAR-Dundee Ltd.,
STAR House, 166 Nethergate
Dundee, DD1 4EE, UK
stuart.mills@star-dundee.com

Steve Parkes
STAR-Dundee Ltd.,
STAR House, 166 Nethergate
Dundee, DD1 4EE, UK
steve.parkes@star-dundee.com

***Abstract*—SpaceFibre (ECSS-E-ST-50-11C) is the latest generation of spacecraft onboard data-handling networks, evolving from the widely adopted SpaceWire standard to meet the increasing demands for higher data rates, enhanced reliability, and robust quality-of-service (QoS) capabilities. SpaceFibre reached TRL-9 in April 2021. Recent developments have further enhanced the capabilities of SpaceFibre, addressing challenges associated with advanced space missions.**

A higher-speed variant of SpaceFibre using 64b/66b encoding surpasses current solutions based on 8b/10b encoding, which can achieve lane rates in the order of 10 Gbit/s, resulting in 40 Gbit/s link rates on a quad-lane link. The novel and efficient encoding scheme enables lane speeds of 25 Gbit/s in current space-qualified technology with minimal protocol stack modifications. An initial implementation demonstrated aggregate data rates of 100 Gbit/s using a quad-lane configuration and was successfully validated on an AMD Versal FPGA during heavy-ion radiation testing campaign.

To support these high speeds efficiently, a SpaceFibre Processor Endpoint has been developed to integrate SpaceFibre networks with processor-based System-on-Chips (SoCs) running operating systems such as Linux. Performance results confirm the effectiveness of this integration strategy, demonstrating substantial benefits for representative spacecraft data-handling systems. This hardware-accelerated endpoint interface achieves data rates into user-space memory at tens of Gbit/s with minimal processor overhead.

This paper presents the new higher-speed variant of SpaceFibre. Then introduces the SpaceFibre Processor Endpoint and software, demonstrates different implementations, and analyses performance results. Collectively, these advancements significantly enhance the throughput, efficiency, and integration flexibility of SpaceFibre. They open new possibilities for spacecraft data-handling architectures, enabling unprecedented levels of performance and reliability.

TABLE OF CONTENTS

1. INTRODUCTION

Emerging spacecraft instruments and on-board processing chains require reliable multi-tens-of-gigabit data links with low latency. While these speeds are currently supported by commercial off-the-self (COTS) standards such as PCI Express (PCIe) and InfiniBand, supporting these speeds in space missions is challenging due to the specifics of the space environment, such as the presence of ionizing radiation, strict power limitations, and wide thermal variations. In parallel, the use of FPGAs in space designs is commonplace thanks to the high flexibility provided by such components. However, this comes with its own challenges, as FPGAs are slower than dedicated ASICs. For all these reasons, high-speed protocols optimized for space applications such as SpaceFibre (SpFi) [1], which provides deterministic quality of service (QoS) and fault detection, isolation, and recovery (FDIR), are

becoming essential. The current version of the SpFi standard supports speeds limited to lane rates around 10 Gbit/s—or aggregate link rates in the order of 50 Gbit/s in multi-lane configurations. However, meeting higher speed requirements on existing radiation-tolerant FPGAs is challenging due to coding overheads, clock-data-recovery limits at high speed, and host-processor bottlenecks. This work presents two tightly coupled contributions aimed at enabling very-high-speed SpaceFibre operation on space-qualified devices while making that performance available to processors.

Firstly, a new higher-speed SpaceFibre variant is presented. The legacy 8b/10b encoding is replaced with 64b/66b, which reduces line-coding overhead while preserving most of the SpaceFibre functional behavior. A wider data path, per-control-word cyclic redundancy checks (CRC) and provision for forward-error correction (FEC) support integrity and scalability at elevated lane rates on existing hardware. The architecture doubles the internal data path width, easing timing closure in radiation-tolerant FPGAs and enabling 25 Gbit/s per lane on the radiation-tolerant Versal device.

Secondly, a Processor Endpoint for SpaceFibre is presented. This hardware-software stack couples SpaceFibre to processor-based systems-on-chip (SoCs) and hosts via hardware queues, zero-copy I/O, and low-latency broadcast/time-distribution primitives. The design sustains throughput of tens of Gbit/s to user-space memory with minimal CPU utilization.

This paper is organized as follows. Section 2 describes the design and implementation of the high-speed SpaceFibre variant. Section 3 introduces the SpaceFibre Processor Endpoint, its architecture, and its integration with processor-based systems. Finally, Section 4 presents the conclusions and outlines future work.

2. HIGHER-SPEED SPACEFIBRE

New spacecraft missions are being designed to generate and process data at ever-increasing rates. Achieving throughputs in excess of 100 Gbit/s per link is currently possible with the in-built transceivers of the Versal family of radiation-tolerant FPGAs, featuring 26 Gbit/s per channel transceivers—and up to 112 Gbit/s in existing commercial variants. This creates an opportunity to improve SpFi to take advantage of the new hardware. Consequently, a modification of the SpFi protocol encoding layer has been developed, aiming to support per-lane data rates of up to 25 Gbit/s in Versal [2], while also preparing for future radiation tolerant devices offering channel rates exceeding 100 Gbit/s, following the path of existing commercial parts.

Lane Encoding and Transcoder

The 8b/10b coding scheme used by SpFi imposes a 20% overhead and limits achievable lane rates, despite providing advantages such as strict DC balance, invalid code detection and symbol synchronization. The proposed update replaces 8b/10b with 64b/66b encoding, which uses a 2-bit synchronization header distinguishing between data and control blocks. The 64b/66b overhead of 3.125% yields a significant efficiency gain compared to 8b/10b. This new encoding is statistically DC-balanced and exhibits a spectrum closer to white noise, providing better transition statistics and spectral properties, thereby improving data recovery and adaptive equalization [3]. The updated encoding is intended for operation at lane rates of 10 Gbit/s or higher.

The transcoder block converts the original 8b/10b coded SpFi data and control words into 66-bit data and control blocks. In 64b/66b encoding, a 64-bit word is transformed into 66 bits through the addition of a 2-bit header, either “10” or “01”. This guarantees a transition at least every 66 bits, effectively limiting the maximum run length. Furthermore, the short block size minimizes latency and allows fast recovery in the event of alignment loss, which can be caused by radiation.

In transmission, the transcoder accepts either two 32-bit data words or a single control word. This means that the link layer must be modified to generate data frames with an even number of data words. The transcoding rules are designed for optimal timing performance and minimal logic and power consumption. They exploit the constraints on the potential values of 8b/10b codes determined by valid SpFi control and data words. A detailed description of the transcode operation has been previously presented [2].

Two consecutive 32-bit data words are mapped into a single data block, while control blocks carry a 32-bit control word followed by a 32-bit CRC. This mapping rule avoids large lookup tables and shortens combinational paths, facilitating timing closure on radiation-tolerant devices. By doubling the number of data words processed per cycle while maintaining the same rate of control-word processing, the approach imitates that used in the SpFi multi-lane configuration: it adds support for higher data rates while maintaining the core clock frequency.

Figure 1 shows a functional block diagram illustrating the proposed changes to the SpFi protocol.

Error Detection and Scrambling

Integrity protection is provided by a 32-bit CRC appended to each control word, covering all data words transmitted since the previous CRC. This mechanism detects multi-bit and burst errors prior to higher-layer processing, complementing higher-level FDIR mechanisms [4]. The resource cost of the CRC generator is minimal, and throughput is not affected.

A self-synchronizing scrambler randomizes payloads continuously, without reinitialization, to reduce spectral peaks, limit run lengths and assist equalizer performance. Synchronization headers remain unscrambled, ensuring fast and reliable block detection and limiting the maximum run length to one block.

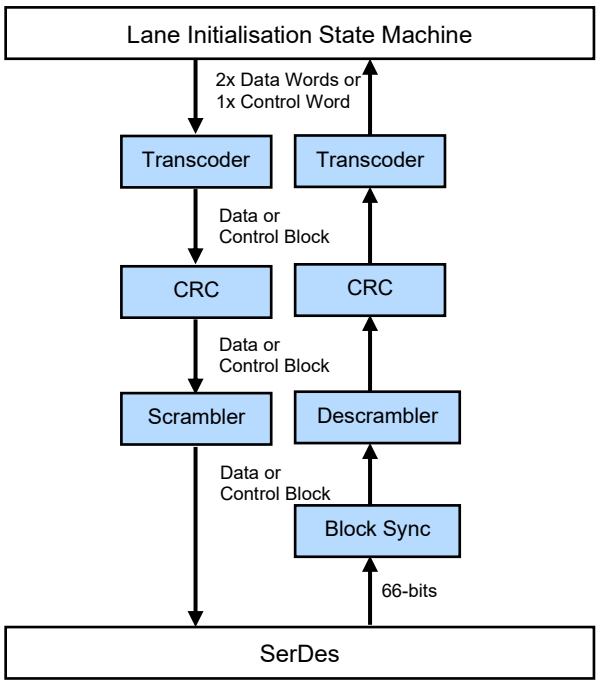


Figure 1. SpFi New Encoding Layer With 64b/66b Encoding

Block Synchronization and Recovery

The receiver achieves block synchronization using the 66-bit block headers "10" and "01". It declares lock established after detecting a predefined run of 64 valid headers and declares lock lost after exceeding an error threshold of 16. This strategy provides both rapid acquisition and strong immunity to isolated bit errors. Following a radiation-induced upset, the receiver automatically reacquires synchronization within a few microseconds, without higher-layer intervention.

Forward Error Correction

Although the new variant currently relies on CRC detection, at lane rates beyond 25 Gbit/s the use of very high frequencies and advanced channel modulations such as PAM4 may require forward-error correction (FEC) techniques. Ongoing work is evaluating different FEC schemes to reliably support operation beyond the 25 Gbit/s threshold.

Prototyping and Validation

The new design has been implemented in AMD Versal (Versal XQRVC1902) [5] and Microchip PolarFire FPGAs (i.e., RTPF500ZT) [6]. Results in Table 1 demonstrate that the new encoding occupies only a small fraction of available logic resources and that timing closure is not affected by this addition. The two-words-per-cycle datapath reduces the required core-clock frequency by a factor of two compared with the 8b10b version, providing comfortable timing margins at 25 Gbit/s per lane for a high-performance FPGA such as Versal. In PolarFire, operation at the maximum supported transceiver speed (12.7 Gbit/s) has also been verified.

Table 1. Resource Usage of the SpFi 64b/66b Encoding Block

	DFF	*LUT*	*RAM*
XQRVC1902	899 0.05%	704 0.08%	0
RTPF500ZT	312 0.07%	562 0.11%	0

In a quad-lane configuration, the design successfully achieved an aggregate data rate of 100 Gbit/s (25 Gbit/s per lane) using the AMD VCK190 Evaluation Kit. It was subsequently tested under heavy-ion irradiation (Figure 2), validating the new encoding in more representative conditions for space applications [7].

Figure 2. Test Setup with the VCK190 Board in Place for Radiation Testing

3. SpaceFibre Processor Endpoint

Increasing link rates requires efficient integration with processor subsystems. Otherwise, higher throughput cannot be effectively utilized if host systems cannot process the resulting traffic. Conventional software interfaces often incur high overheads from copy operations and interrupts, limiting throughput as processor utilization quickly becomes a bottleneck. The Processor Endpoint was therefore developed to provide a hardware-accelerated interface between SpaceFibre networks and processors, with the goal of maximizing performance while keeping processor overhead minimal. It builds on previous work on Remote Direct Memory Access (RDMA) over SpFi implemented on a Zynq UltraScale+ MPSoC. This earlier work demonstrated user

data transfers at 23 Gbit/s with less than 1% processor utilization [8].

The objectives of the SpFi Processor Endpoint IP are to:

- Deliver sustained user-space throughput in the tens of gigabits per second;
- Minimize CPU utilization by avoiding redundant copies and reducing interrupt frequency;
- Preserve software support for low-latency SpFi broadcast messages, as well as compatibility with SpaceWire (SpW) interfaces;
- Support integration both in system-on-chip (SoC) architectures and in remote processor systems using discrete FPGA accelerators connected via a high-speed interconnect such as PCIe.

Architecture Overview

The Processor Endpoint follows a layered design, as depicted in Figure 3. The bottom layer consists of a configurable number of SpW ports and SpFi virtual channels (VCs). These VCs can reside within one or several SpFi ports. The next layer corresponds to the SpFi Processor Endpoint IP, which interfaces the ports and VCs in the layer below. These interfaces allow the transfer of packets, time-codes (SpW) and broadcast messages (SpFi) between the Endpoint IP and the SpW/SpFi network. The kernel-space drivers manage communication between the host processor and the SpFi Processor Endpoint IP, establishing the path between the user-space APIs and the Endpoint IP. Finally, the user-space application programming interfaces (APIs) provide the interface between user applications and the kernel-space drivers.

Virtual channels are mapped directly to software queues, preserving end-to-end QoS. Broadcast and time-code primitives are supported in hardware to ensure low-latency behavior.

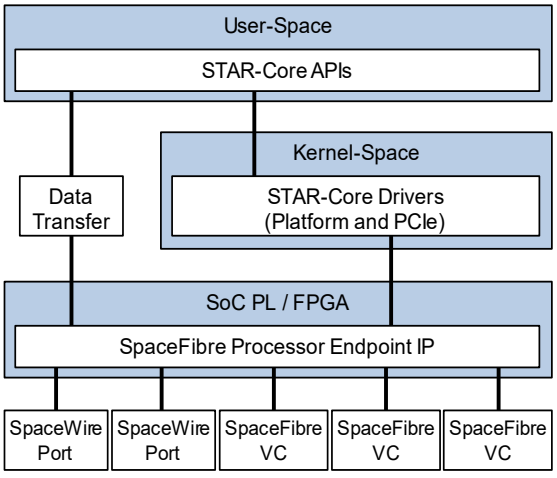


Figure 3. SpaceFibre Processor Endpoint Architecture

Integration with Processor Systems

Two integration architectures have been demonstrated:

- System-on-Chip (SoC): The Processor Endpoint is instantiated in the programmable logic of devices such as Zynq UltraScale+ [9] and Versal [5] (Figure 4). The processor subsystem runs Linux, with the driver bound through the device tree. This configuration is suited to payload controllers that combine communication with embedded processing.
- Discrete FPGA: The Processor Endpoint is deployed in a standalone FPGA (e.g. Microchip PolarFire SoC [10]) connected to an external processor running Linux via a high-speed interconnect such as PCIe (Figure 5). The host executes the Linux driver and user-space applications. This approach suits systems where powerful general-purpose processors must be coupled to radiation-tolerant network interfaces.

Both approaches provide the same API to the user, simplifying portability across platforms. A bare-metal software option is also being developed for cases where a full operating system is not required.

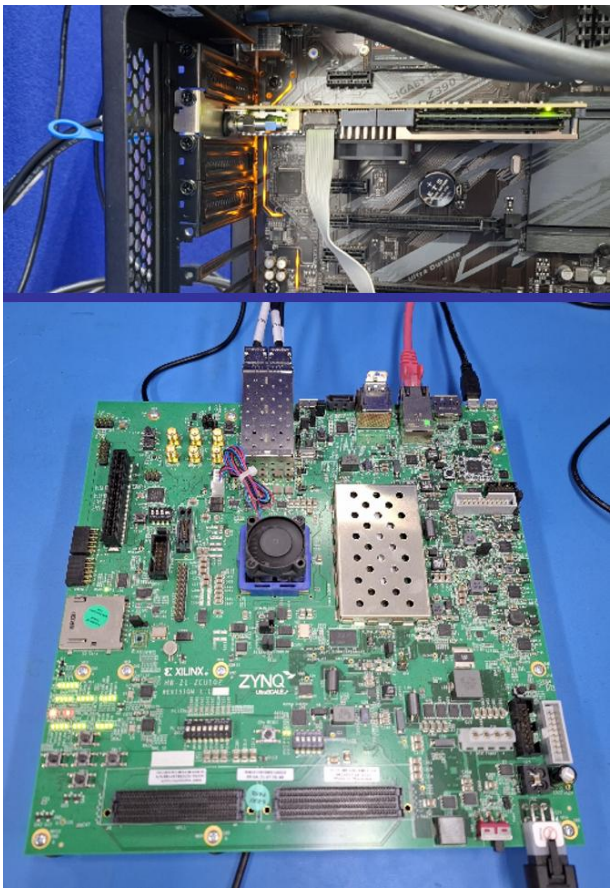


Figure 4. Fig. 2. SpaceFibre Processor Endpoint on AMD Zynq UltraScale+ MPSoC (ZCU102)

Figure 5. SpaceFibre Processor Endpoint on Microchip PolarFire

Performance Evaluation

The performance test setup consisted of a host PC and a device under test connected via a quad-lane SpaceFibre link. The host PC was an Intel i9-9900K machine with 128 GB DDR4 memory and NVMe storage, running Windows 11 Pro. A STAR-Ultra PCIe board provided the SpaceFibre interface and a PCIe Gen3 x8 host interface [11], with STAR-System drivers and APIs used to generate and receive test traffic [12]. The device under test was an AMD Zynq UltraScale+ MPSoC (ZCU102 board), running Linux on its ARM Cortex-A53 cores in the Processing System (PS), with the SpaceFibre Processor Endpoint IP implemented in the Programmable Logic (PL).

The boards were interconnected through a quad-lane link operating at a lane signalling rate of 7.5 Gbit/s per lane, yielding an aggregate link rate of 30 Gbit/s. This configuration corresponded to a maximum unidirectional throughput of approximately 22.9 Gbit/s and a maximum bidirectional throughput of about 22.2 Gbit/s. The setup enabled controlled measurement of data rates and processor utilization across varying packet sizes and transmission strategies.

The performance test procedures evaluated different send and receive strategies for the SpaceFibre Processor Endpoint. Specifically, four communication scenarios were defined: sending single or multiple packets from the Endpoint to the STAR-Ultra PCIe board, and receiving single or multiple packets. For each scenario, packet sizes ranged from 1 kB to 256 kB, increasing in powers of two. Each packet size was tested in 10 iterations, each lasting 10 seconds, to obtain statistically representative results. During the tests, throughput was measured by counting the total number of bytes transmitted or received at the Endpoint, while CPU utilization was obtained from Linux system statistics.

Table 2 presents the transmission results. A peak throughput exceeding 20 Gbit/s was achieved with CPU utilization below 3%. Batch transmission further reduced CPU overhead to less than 1% while sustaining comparable throughput.

Table 3 presents the reception results. Immediate per-packet completion scaled to approximately 20 Gbit/s with around 2% CPU utilization. Threshold-based batching significantly improved efficiency for small packets while maintaining high throughput for large transfers.

These results demonstrate that the Processor Endpoint can sustain tens of gigabits per second of user-space throughput on representative SoC platforms with minimal processor load.

Table 2. Sending Single and Multiple Packets Results

	Sending Single Packets		Sending Multiple Packets	
Test Case	Data Rate (Gbit/s)	CPU Use (%)	Data Rate (Gbit/s)	CPU Use (%)
1 KB	1.57	25.06	13.83	4.94
2 KB	3.16	25.09	19.19	3.79
4 KB	4.87	24.86	21.19	2.15
8 KB	5.70	15.68	22.01	0.51
16 KB	8.13	13.80	22.22	0.71
32 KB	13.14	11.48	22.31	0.73
64 KB	17.07	7.09	22.36	0.56
128 KB	19.55	4.70	22.39	0.79
256 KB	21.09	2.49	22.41	0.68

Table 3. Receiving Single and Multiple Packets Results

	Receiving Single Packets		Receiving Multiple Packets	
Test Case	Data Rate (Gbit/s)	CPU Use (%)	Data Rate (Gbit/s)	CPU Use (%)
1 KB	2.20	22.82	11.96	3.81
2 KB	4.40	22.82	13.49	2.98
4 KB	8.80	22.81	13.61	2.72
8 KB	14.77	22.79	15.42	1.83
16 KB	17.04	18.89	17.49	1.72
32 KB	18.47	13.95	18.76	1.56
64 KB	19.27	7.72	19.46	1.59
128 KB	19.69	3.85	19.83	1.84
256 KB	19.92	1.94	20.02	1.27

4. CONCLUSION

The higher-speed SpaceFibre physical layer and the SpFi Processor Endpoint both address complementary aspects of the same challenge: delivering very high-speed and reliable communication in modern spacecraft data-handling systems implemented on radiation-tolerant FPGAs with embedded or external processors.

The encoding layer redesign replaces 8b/10b with 64b/66b encoding and introduces a new 32-bit CRC for each control word. These modifications reduce overhead and provide improved integrity checks while preserving the main SpaceFibre semantics. Operation at 25 Gbit/s per lane has been demonstrated, with an aggregate throughput of 100

Gbit/s across four lanes. Heavy-ion testing validated the initial implementation under radiation conditions.

The Processor Endpoint complements this advance by enabling processors to sustain high throughput without incurring excessive CPU overhead. Hardware queues, zero-copy interfaces, and virtual-channel mapping ensure that tens of gigabits per second can be transferred between user-space memory and the SpFi network. Results from a representative SoC platform confirm the practicality of integrating high-speed SpFi links into processor-based architectures.

Together, these contributions demonstrate that 100 Gbit/s-class, radiation-robust networks can be realized with current flight-qualified FPGAs. The approach provides clear a migration path to support higher lane rates and ensures that host processors can exploit the resulting link capacities efficiently.

Future work will focus on end-to-end system validation and FEC integration for operation beyond 25 Gbit/s per lane.

REFERENCES

[1] ECSS Standard ECSS-E-ST-50-11C, “SpaceFibre – Very high-speed serial link”, European Cooperation for Space Data Standardization, 15th May 2019, available from http://www.ecss.nl.

[2] A. Ferrer, A. Gonzalez, M. Farras and S. Parkes, "100 Gbit/s-plus SpaceFibre on Space-Qualified FPGAs," *2025 European Data Handling & Data Processing Conference (EDHPC)*, Elche, Spain, 2025.

[3] Texas Instruments, “System Design Considerations when Upgrading from JESD204B to JESD204C”, Application Report SBAA402A, February 2019 (revised September 2025).

[4] Szczepanek, "IEEE 802.3ap MTTFPA Calculations", 2005.

[5] AMD, “XQR Versal for Space 2.0 Applications Product Brief”, AMD, 2023, available from https://www.xilinx.com/content/dam/xilinx/publications/product-briefs/xilinx-xqr-versal-product-brief.pdf.

[6] Microchip, “DS5395 Radiation-Tolerant PolarFire® SoC FPGA Product Overview”, Microchip, 2024, available from https://ww1.microchip.com/downloads/aemDocuments/documents/FPGA/ProductDocuments/ProductBrief/Radiation-Tolerant-PolarFire-SoC-FPGA-Product-Overview-DS00005395.

[7] A. Gonzalez, A. Ferrer, M. Farras, S. Parkes, P. Maillard and K. O'Neill, "Characterisation of AMD Versal FPGA Transceivers Under Heavy-Ion Radiation," 2025 25th European Conference on Radiation and Its Effects on Components and Systems (RADECS), Antwerp, Belgium, 2025.

[8] D. Gibson, S. Mills, A. MacLennan and S. Parkes, “Efficient High Data Rate Networking Using Remote Direct Memory Access Over SpaceFibre”, European Data Handling & Data Processing Conference (EDHPC), Juan Les Pins, France, October 2023.

[9] AMD, “Zynq UltraScale+ MPSoCs”, https://www.amd.com/en/products/adaptive-socs-and-fpgas/soc/zynq-ultrascale-plus-mpsoc.html.

[10] Microchip, “PolarFire SoC FPGAs”, https://www.microchip.com/en-us/products/fpgas-and-plds/system-on-chip-fpgas/polarfire-soc-fpgas.

[11] S. Parkes, D. Gibson, S. Mudie, C. McClements, A. Ferrer Florit and A. Gonzalez Villafranca, “Testing SpaceFibre Interfaces and Systems”, Data Systems in Aerospace (DASIA) Conference, Online, September 2021.

[12] S. Mills and S. Parkes, “A Software Suite for Testing SpaceWire Devices and Networks”, Proceedings of Data Systems in Aerospace (DASIA) Conference, Barcelona, Spain, 2015.

BIOGRAPHY

Alberto Gonzalez Villafranca *holds a PhD in data compression for space applications and has been connected to the space field throughout his entire professional career. He began by collaborating with the Gaia mission and subsequently worked on the hardware implementation of a deterministic variant of the SpaceWire protocol at the European Space Agency. Since then, Alberto has been deeply involved in the definition and implementation of SpaceFibre for more than 10 years, first with STAR-Dundee Ltd and later with its subsidiary, STAR-Barcelona, as a Chip Designer.*

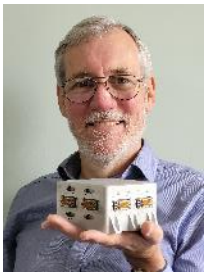
Albert Ferrer Florit *has a PhD in high-speed interconnection networks for space applications awarded by the University of Dundee. His PhD research was funded by ESA's Networking/ Partnering Initiative after he worked in the on-board data processing group (TEC-EDP) in ESTEC. He is specialised in SpaceWire and SpaceFibre networks, being one of the key developers of the SpaceFibre standard. He started his career at CERN in the Summer Student Programme, worked for STAR-Dundee and is currently working for STAR-Barcelona as a Network and FPGA Engineer.*

Marti Farras Casas *holds a BSc. in Electronics and already began working in the space industry while at university, first with an internship on ESA's Gaia mission, followed by his degree thesis developed at STAR-Dundee Ltd. For over eight years, Marti has been deeply involved with the development of SpaceFibre, first with STAR-Dundee Ltd and later with its subsidiary, STAR-Barcelona, as a Hardware Design Engineer.*

Dave Gibson *received a BSc. in Applied Computing and a PhD from the University of Dundee, where he focused on SpaceWire-D, a deterministic protocol running on top of existing SpaceWire networks and technology. Since 2015, he has been a software engineer at STAR-Dundee working on a variety of SpaceWire and SpaceFibre applications including Remote Direct Memory Access over SpaceFibre, SpaceWire and SpaceFibre software for embedded systems, network monitoring and control software, and test and development equipment.*

Andrew MacLennan *received a Joint BEng in Computer Science and Electronics from the University of Edinburgh. He initially worked on embedded software for efficient display systems on mobile devices, before moving into academic research and development of power-efficient video encoders. Before joining STAR-Dundee he was a system architecture designer for FPGA-based high-performance video encoder systems for the broadcast industry. Andrew is an FPGA Engineer at STAR-Dundee, where he has worked on several SpaceFibre applications, including a 10 Gbit Ethernet to SpaceFibre bridge, Remote Direct Memory Access over SpaceFibre, a SpaceFibre Router test and development product, and a SpaceFibre / SpaceWire endpoint for processor-based systems.*

Stuart Mills *received a BSc. in Computer Science and a PhD from the University of Dundee, where he focused on higher layer protocols for SpaceWire and the design of high throughput and low latency software for interfacing to SpaceWire networks. Prior to starting his PhD, he worked as a software consultant, developing software solutions for several large multinational companies. Stuart is one of the founders of STAR-Dundee and has led the company as Chief Executive Officer since 2017. He has now been working on SpaceWire and SpaceFibre technology for over 25 years.*



Steve Parkes *received a BSc. and MSc. in Applied Physics and Electronics from Lancaster University, and a PhD from University College Cardiff. He is a Fellow of the Royal Academy of Engineering. He worked on Underwater Acoustic Systems for six years and at BAE Space Systems for seven years before moving to Academia. He worked at Dundee University for 24 years where he became Professor of Spacecraft Electronic Systems carrying out research on spacecraft onboard networks, vision-based navigation and related planet surface simulation, and signal processing. He is the author of the widely used ECSS SpaceWire and SpaceFibre standards, with inputs from international engineers. In 2002 he founded STAR-Dundee, which supports industry and space agencies using SpaceWire and SpaceFibre technology, where he is now Chief Technology Officer.*