100 Gbit/s-plus SpaceFibre on Space-Qualified FPGAs

Albert Ferrer Florit
STAR-Barcelona S.L.
Barcelona, Spain
alberto.ferrer@star-dundee.com

Alberto Gonzalez Villafranca
STAR-Barcelona S.L.
Barcelona, Spain
alberto.gonzalez@star-dundee.com

Marti Farras Casas
STAR-Barcelona S.L.
Barcelona, Spain
marti.farras@star-dundee.com

Steve Parkes
STAR-Dundee Ltd
Dundee, United Kingdom
steve.parkes@star-dundee.com

***Abstract*— SpaceFibre (ECSS-E-ST-50-11C) is an advanced spacecraft onboard data-handling network technology. It builds upon its previous generation, SpaceWire (ECSS-E-ST-50-12C), to meet the increasing demands for higher data rates and enhanced reliability in space applications. Increasingly complex payloads and advanced scientific instruments onboard spacecraft may demand data rates well beyond the practical limits of current SpaceFibre solutions using 8b/10b encoding, whose inherent overhead reduces effective throughput compared to more efficient coding schemes.**

This paper presents a new development of the SpaceFibre protocol that enables significantly higher lane rates using a more efficient encoding scheme, requiring minimal modifications to other elements of the protocol stack. The solution implements a transcoding block that efficiently maps SpaceFibre 8b/10b data and K-codes into 64b/66b data and control blocks. Error detection is enhanced with an additional 32-bit CRC. An implementation achieving an aggregate data rate of 100 Gbit/s using a four-lane configuration has already been successfully tested under heavy-ion radiation on a Versal FPGA.

Keywords—SpaceFibre, SpFi, Onboard Networking, Onboard Data Handling, Versal, FPGA

I. Introduction

High-speed communication protocols have evolved significantly over the last decade to address continuously increasing bandwidth demands across various applications. Standards such as PCIe and InfiniBand have progressively increased lane rates, adopting more efficient encoding schemes and advanced modulation to achieve throughput levels well beyond 25 Gbit/s. Space applications, however, have historically lagged behind due to the stringent reliability requirements imposed by harsh radiation environments and constraints on available radiation-tolerant technologies.

Emerging spacecraft missions incorporating increasingly complex payloads, such as high-resolution imaging sensors, radar systems, and scientific instruments, require significantly higher data transfer rates. With the emergence of advanced FPGAs such as the Versal series, which support transceiver speeds of up to 25 Gbit/s in space-qualified devices (and up to 112 Gbit/s in commercial versions), the need arises to improve SpaceFibre's (SpFi) encoding efficiency to take full advantage of modern hardware and meet upcoming mission requirements [1].

With higher space-qualified lane rates anticipated in the near future, this work explores techniques to substantially boost SpaceFibre data throughput while maintaining reliability under both nominal Bit Error Rate (BER) and radiation events.

II. Trade-Offs and Design Drivers

Support for higher data rates should be achieved with minimal changes to the SpaceFibre standard so that existing implementations can be adapted to operate at both legacy and enhanced speeds while preserving backward compatibility.

The main modification concerns the encoding layer defined in the standard. The design objective is to identify an encoding scheme supported by modern FPGA transceivers that is better suited for high-speed operation. If the selected encoding lacks certain functions of the current scheme, these should be implemented in additional logic to ensure that the complete solution maintains comparable or improved capabilities.

The solution must also remain suitable for implementation as a soft IP core on radiation-tolerant FPGAs, for which achieving very high data rates poses significant implementation challenges.

A. Line Encoding

Current SpaceFibre implementations employ 8b/10b encoding, a robust scheme that provides strict DC balance, disparity and invalid code detection, symbol synchronization and control codes, but with a 20% overhead.

While strict DC balance limits baseline wander, scrambling-based schemes generate bit streams that are statistically DC balanced and spectrally closer to white noise, providing stable, near-random transition statistics and more uniform spectral properties. These features are especially advantageous at higher data rates, where they facilitate clock/data recovery and adaptive equalization (CTLE, DFE, etc.) [2].

Encoding methods such as 64b/66b and 128b/130b employ scrambling and reduce encoding overhead down to 3.125% and 1.538%, respectively [3]. These two schemes employ two additional bits per codeword that limit the run length, provide word alignment and allow distinction between data and control symbols.

Further overhead reductions, as in the case of 256b/257b encoding, remove explicit alignment bits altogether by relying on periodic known DC-balanced scrambled patterns for word alignment. Additionally, the PCIe Flit approach eliminates the dedicated bit for distinguishing data and control, as positional

or temporal fixed structures implicitly manage these distinctions [4].

Among the considered encoding methods, 64b/66b was ultimately selected due to an optimal trade-off between encoding efficiency, alignment complexity, latency, and hardware availability. Specifically:

- 64b/66b dramatically reduces encoding overhead compared to 8b/10b, from 20% down to 3.125%, significantly improving throughput.
- Shorter block size with explicit header bits for alignment and data/control separation, minimizes latency and ensures rapid recovery from alignment loss, essential in radiation environments where events such as PLL Single Event Effects (SEEs) cause transient alignment loss.
- Since the 2-bit sync header is always “01” or “10” and not scrambled, each block contains a guaranteed transition, limiting the maximum run length to 66 bits.
- Most modern FPGA transceiver hard-IP blocks, including radiation-tolerant variants, natively support 64b/66b encoding. Utilizing FPGA hard-IPs reduces area overhead and simplifies timing closure.

Additionally, the 64-bit payload of the 64b/66b block is a natural fit for mapping SpaceFibre word formats and for accommodating new error-detection functionality. As discussed throughout this work, a single block carries either two 32-bit SpaceFibre data words or one 32-bit control word followed by a 32-bit checksum.

B. *Error Detection*

The 8b/10b encoding inherently detects all single-bit errors and a subset of multi-bit errors, whereas 64b/66b provides no built-in error detection capability. To address this limitation, and given the higher bit error rates expected at increased lane rates, an additional integrity check is required. A suitable approach is to append a 32-bit CRC immediately after each control word, covering all transmitted bits since the preceding CRC. This ensures the integrity of both SpaceFibre frames and control words before they are processed.

C. *Efficient FPGA implementation*

Current SpaceFibre IP cores typically process one control word per clock cycle, while the number of data words processed per cycle depends on the number of lanes in the link. To support higher lane rates without increasing the target clock frequency, the processing rate of control words should remain unchanged, and instead the number of data words per cycle should be increased, as is already done in multilane configurations. With 64b/66b encoding, a suitable approach is to process, in each clock cycle, either two 32-bit SpaceFibre data words or one 32-bit control word followed by a 32-bit CRC. This effectively doubles the throughput achievable at the same target frequency, enabling efficient FPGA implementations at higher data rates.

Regarding the mapping of the 8b/10b symbols used by the SpaceFibre data and control words into the 64b/66b blocks, a simple set of translation rules is preferred, rather than a lookup table (as in 10G Ethernet)[5], with the rules themselves defined to require the minimum possible amount of logic.

III. SPECIFICATION

This section describes the specifications for SpaceFibre lanes operating at 10 Gbps or more. At these speeds, the SpaceFibre lane layer logic related to 8b/10b encoding, symbol and word synchronisation is replaced by the new data flow based on 64b/66b encoding, as shown in Figure 1.

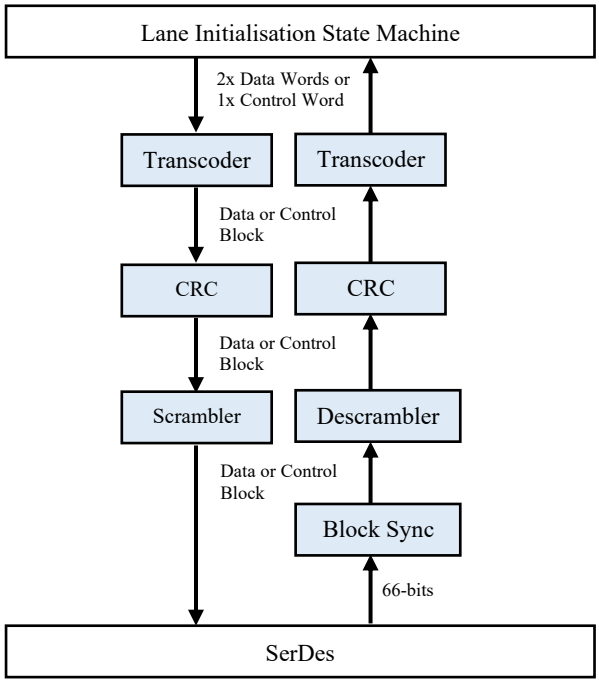

Fig. 1. Functional block diagram with 64b/66b encoding.

A. *Transcoder block*

The transcoder block converts the 8b/10b codes of the SpFi data and control words into 66-bit data and control blocks. In transmission, the transcoder only 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. It exploits the constraints on the potential values of 8b/10b codes determined by valid SpFi words.

The main rule is that if a SpFi word contains one control character (i.e. 8b/10b K-Code) then the word is encoded in a control block. Therefore, a control block can contain either a control word or two data words with at least one EOP/EEP or FILL control character. All control words are mapped to control blocks. Data words can be mapped to data or control blocks. Figure 2 shows this mapping.

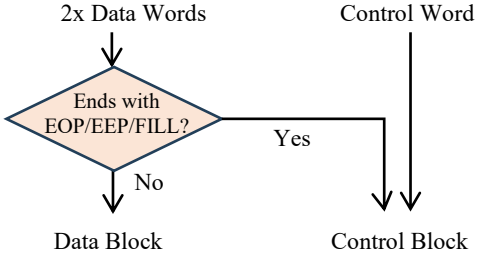

Fig. 2. Mapping of SpFi words to data and control blocks.

The actual mapping is straightforward for data blocks and for control blocks containing a control word. In these cases, the block contains just the data bytes of the data or control word. In the case of data words with control characters, it differs in that the first data byte is placed in the last byte of the control block and the first byte of the control block contains information about the control characters.

Figure 3 shows the detailed mapping. The 64-bit data bytes of the two data words are placed in the data block. The 32-bit data bytes of the control word are placed in the lowest significant bit positions of the control block, followed by a 32-bit CRC field. When the data word ends with EOP/EEP/FILL then the first byte "C" contains the first two bits of the last data byte of the data word, followed by the K-Flags from byte 1 to 6. The last byte contains the first byte of the data word instead of the data byte value of the EOP/EEP/FILL.

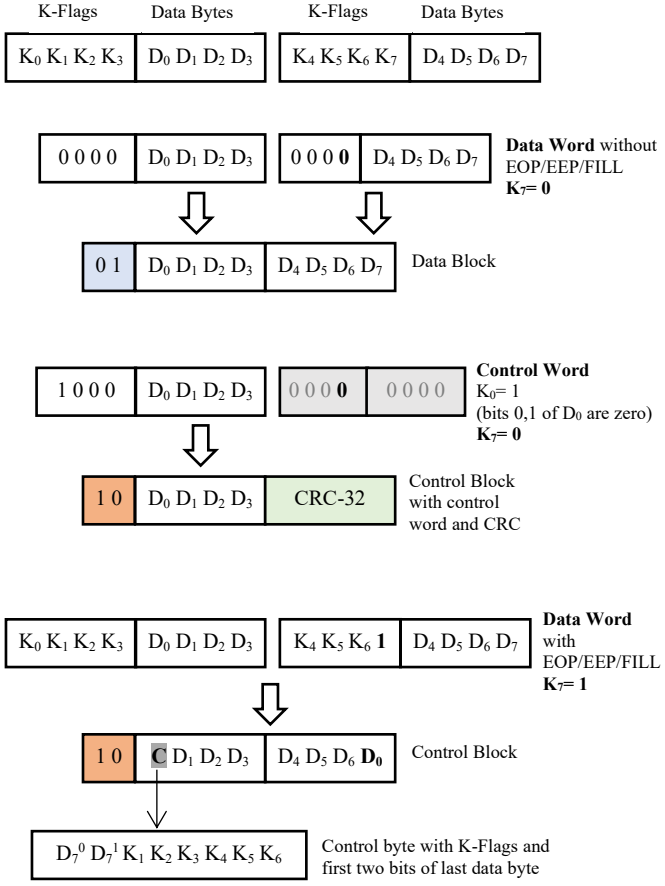

Fig. 3. Mapping between SpFi words into data and control blocks.

When a control block is received, Table 1 shows how it can be decoded into control or data words depending on the first two bits of the first byte. If their values are "00" then the control block contains a control word, which has specific K-Flag values. Otherwise, the control block contains two data words with its last character "D_7, K_7" being an EOP, EEP, or FILL control character. The first byte C does not provide the first K-Flag (K_0) value, which must be derived from "D_1, K_1" using the expression "$K_0 = K_1 \,\&\, D_1^0 \,\&\, D_1^1$". This means that the first character must be an EOP, EEP or FILL (i.e. K_0 is "1") when the second byte is a FILL character.

TABLE I. SPACEFIBRE CONTROL CHARACTERS

Control Character Type (K-Flag = 1)	Data byte	
	Bit 0	*Bit 1*
First control character of a control word	0	0
EOP	0	1
EEP	1	0
FILL	1	1

B. CRC block

The CRC-32 polynomial defined in IEEE 802.3 is used to generate the 32-bit CRC field appended to each control block that contains a SpFi control word. The CRC-32 guarantees detection of all one-bit, two-bit, and three-bit errors, all burst errors up to 32 bits in length, and any pair of burst errors up to eight bits each [6]. In our implementation, this CRC covers the 64-bit payload of every 66-bit block transmitted since the preceding CRC field, including the current control word, and is therefore re-initialized after each control word. This mechanism ensures that errors affecting control words, data frames, or broadcast frames are detected prior to higher-layer processing. For example, if an error occurs within a data frame, it will be detected by the CRC included in the control block that contains the end of data frame control word (i.e., EDF). The CRC does not cover the 2-bit sync header. An error causing a data block to be misinterpreted as a control block will yield a random CRC value and will be detected, while an error causing a control block to be misinterpreted as a data block will be detected by the CRC appended to the next control word.

C. Scrambler

The self-synchronous scrambler specified for 10GBASE-R (IEEE 802.3, Clause 49) is employed. This scrambler is defined by the polynomial $x^{58}+x^{39}+1$ and operates across the 64-bit payload of each 66-bit block, while leaving the 2-bit sync header unscrambled. The scrambling process runs continuously across consecutive blocks without reinitialization.

D. Block Sync

The block synchronizer aligns the receive data stream to 66-bit boundaries by detecting the unique sync header patterns "01" (data block) and "10" (control block). After a sufficient number of consecutive correct headers are observed, block lock is declared and the receiver thereafter uses the header bits both for alignment maintenance and for distinguishing between data and control blocks. Loss of block lock is declared if the number of header errors within a defined observation window exceeds a threshold. In our implementation, we adopt the same thresholds as IEEE 802.3 Clause 49: block lock is declared after 64 consecutive valid sync headers, and loss of block lock is declared if 16 or more invalid headers occur within any 64-block window.

IV. PROTOTYPING AND VALIDATION

A prototype incorporating the proposed 64b/66b-based SpaceFibre implementation was developed and verified using industry-standard simulation methodologies. Randomized and directed test scenarios were employed to stress corner cases and validate alignment, error detection, and recovery mechanisms.

A. Hardware validation

The design was validated in hardware, at 25 Gbit/s per lane, using a VCK190 Evaluation Kit, whose FPGA is functionally equivalent to its radiation-tolerant counterpart (Versal XQRVC1902 [7]). The design was also implemented for PolarFire FPGAs (i.e., RTPF500ZT), to ensure that the design met timing for its maximum transceiver speed of 12.7 Gbit/s [8].

Table 2 reports the resource usage of the new encoding blocks required for 64b/66b encoding, for both the Versal and PolarFire FPGAs. The PolarFire uses even fewer resources because its transceiver implements the scrambler and the block synchronizer logic.

TABLE II. RESOURCE USAGE OF 64B/66B ENCODING BLOCKS

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

B. Radiation Testing on Versal

Subsequent heavy-ion radiation testing was conducted as part of a broader campaign evaluating both the Versal transceivers and the upgraded SpaceFibre link in collaboration with AMD. The experimental setup shown in Figure 4 demonstrated a 100 Gbit/s SpaceFibre link configured with four lanes, each running at 25 Gbit/s [9]

Fig. 4. Test Setup with the VCK190 board in place for radiation testing.

V. CONCLUSION

By updating from 8b/10b encoding to 64b/66b, SpaceFibre can achieve substantially higher data rates, reaching an aggregate of 100 Gbit/s across four lanes in radiation-tolerant FPGAs. This is enabled by a transcoding block that maps SpaceFibre 8b/10b codes into 64b/66b blocks with minimal logic and power overhead.

In contrast to other solutions (e.g., Ethernet, InfiniBand), SpaceFibre does not impose a maximum packet length and provides a reliable link that transparently mitigates radiation-induced SEEs. The ability of SpaceFibre to operate without software intervention is also important when supporting data rates greater than 10 Gbit/s in the constrained hardware environment of space systems.

This work demonstrates that a significant performance increase can be achieved with limited modifications to the existing standard, preserving compatibility while extending SpaceFibre to higher lane rates. Ongoing work focuses on the development of forward error correction (FEC) methods to support reliable lane rates beyond 25 Gbit/s, paving the way for next-generation multi-lane SpaceFibre implementations with aggregate throughputs exceeding 400 Gbit/s.

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] Texas Instruments, “System Design Considerations when Upgrading from JESD204B to JESD204C”, Application Report SBAA402A, February 2019 (revised March 2021)

[3] B. Raahemi, "Error correction on 64/66 bit encoded links," Canadian Conference on Electrical and Computer Engineering, 2005., Saskatoon, SK, Canada, 2005, pp. 412-416, doi: 10.1109/CCECE.2005.1556959.

[4] D. D. Sharma, "PCI Express® 6.0 Specification at 64.0 GT/s with PAM-4 signaling: a low latency, high bandwidth, high reliability and cost-effective interconnect," 2020 IEEE Symposium on High-Performance Interconnects (HOTI), Piscataway, NJ, USA, 2020, pp. 1-8, doi: 10.1109/HOTI51249.2020.00016.

[5] IEEE Computer Society, IEEE Standard for Ethernet—Amendment 1: Media Access Control (MAC) Parameters, Physical Layers, and Management Parameters for 10 Gb/s Operation, IEEE Std 802.3ae™, 2002.

[6] Szczepanek, "IEEE 802.3ap MTTFPA Calculations", 2005

[7] 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 .

[8] 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.pdf .

[9] 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.