




Article

Design of a Reduced SpaceFibre Interface: An Enabling Technology for Low-Cost Spacecraft High-Speed Data-Handling

Gianmarco Dinelli ¹, Pietro Nannipieri ^{1,*}, Daniele Davalle ² and Luca Fanucci ¹

¹ Department of Information Engineering, University of Pisa, 56100 Pisa, Italy; gianmarco.dinelli@ing.unipi.it (G.D.); luca.fanucci@unipi.it (L.F.)

² IngeniArs s.r.l., 56100 Pisa, Italy; danielle.davalle@ingeniars.com

* Correspondence: pietro.nannipieri@ing.unipi.it

Received: 6 May 2019; Accepted: 9 September 2019; Published: 11 September 2019



Abstract: SpaceFibre is an upcoming on-board high-speed communication protocol for space applications. It has been developed in collaboration with the European Space Agency to answer the growing data-rate requirement of satellite payloads such as Synthetic Aperture Radars or hyper-spectral imagers. SpaceFibre offers a complete set of features (i.e., Fault Detection, Isolation and Recovery, and Quality of Service) that guarantees robust communication at the price of higher complexity. This article proposes an innovative modified implementation of the SpaceFibre standard: R-SpaceFibre. It has been designed to reduce hardware resources while keeping high data-rate capability and flow control. Attention is given to the trade-off between Data link layer complexity reduction and protocol features. The proposed protocol is particularly suitable in scenarios where very low bit error rate is foreseen and data integrity is not critical, for example in imaging instruments. The main advantage is a reduction of more than 40% of logical resources required per single interface. R-SpaceFibre may be a suitable solution for several applications, such as low earth orbit CubeSats, which have strict requirements in terms of available logic resources, mass, volume and cost, and more relaxed constraints in terms of upset immunity.

Keywords: SpaceFibre; CubeSats; communication protocol; FPGA; data-handling; hardware optimisation; high-speed

1. Introduction

SpaceFibre is a high data-rate communication protocol specifically designed for on-board data-handling space applications [1]; its standardisation process by the European Cooperation for Space Standardization (ECSS) is at its final steps [2] and is to be published in 2019 (ECSS-E-ST-50-11C). SpaceFibre protocol development has been promoted by the European Space Agency (ESA), which also developed the SpaceWire standard [3], with SpaceFibre representing its evolution. SpaceWire is a full-duplex, low error rate line with low resources utilisation (5–8 Kbytes) [4] and a data-rate up to 400 Mbps. It has already been adopted in several ESA, National Aeronautics and Space Administration (NASA), and Japan Aerospace Exploration Agency (JAXA) missions, such as Rosetta, Mars-Express, Galileo and many others, making it a highly reliable solution. A wide set of CODEC-DECODERS (CODECs) and routers is available on the market, both with Application Specific Integrated Circuit (ASIC) and Field Programmable Gate Array (FPGA) [4], as well as a large choice of support equipment [5]. SpaceWire also implements fault tolerance mechanisms that automatically switches from one link to another in case of failure with very small loss of information. The protocol itself has been designed for easier integration and to improve and promote equipment reuse across different missions, making it also a competitive choice in terms of costs.

SpaceWire adoption by different space agencies is therefore well motivated by the above considerations. Nevertheless, most recent applications require several Gbps data-rate, which cannot be achieved by a single SpaceWire node. In the near future, ESA upcoming missions (i.e., FLEX [6] and BIOMASS [7]) will mount Synthetic Aperture Radars (SARs) and hyper-spectral imagers which will require high speed on-board communication. In current missions, the lack of a standardised high-speed communication protocol led system integrators to adopt not optimised solutions such as WizardLink interfaces (not standardised) [8] with customised protocol on top or more SpaceWire CODECs in parallel (more components, cables and connectors required). These reasons fostered SpaceFibre development, starting from the same basic concepts of SpaceWire and adding functionalities such as greater bandwidth, the concept of Virtual Channels (VCs), Quality of Service (QoS) and Fault Detection Isolation and Recovery (FDIR). In fact, SpaceFibre features meet the data-rate requirement of future missions: a single lane SpaceFibre link can achieve 6.25 Gbps bandwidth, which is necessary to support data-rate requirements of most advanced payloads. Other solutions are currently under evaluation by the space community to be selected as the next standardised high-speed communication protocol for space applications: in particular, RapidIO [9] and TTEthernet [10]. Nevertheless, the adoption of these solutions is not straight forward. In fact, even if they fulfil system requirements, they have several disadvantages: they are not based on an open ECSS standard, and the structure of the protocol itself is not developed aiming at optimisation for spacecraft usage but rather it is an adaptation of commercial technology to space requirements. SpaceFibre instead is based on an open standard and could be easily implemented by different entities, increasing its reliability. Even if the standard has not been released yet, a wide range of SpaceFibre based product is available, both as research project or commercial product, from CODEC Intellectual Property (IP) (from IngeniArs [11], ESA [12] and StarDundee [1]) to routing switch (IngeniArs [13] and StarDundee [14]) and network simulator (IngeniArs [15]). Moreover, it has been developed specifically for space applications, resulting in a highly optimised solution. However, in particular cases, SpaceFibre may also be over-specified. For instance, the data retry feature is not even removable in payloads characterised by streaming of large amount of data which do not contain critical information (i.e., high-resolution imaging payloads, where it is acceptable to lose data frames in case of upsets). Moreover, for small satellites and CubeSats [16], where the size of components particularly matters, both high-speed interface and data processing unit are to be embedded in the same FPGA. An attempt to design a SpaceFibre endpoint with reduced complexity has already been presented in [17], where an architecture to reduce area occupation has been proposed. Unfortunately, the lack of details on hardware configuration (i.e., target frequency, implementation of data scrambling, VCs size, etc.) combined with the fact that the standard significantly developed since 2014 makes it inappropriate to compare analytically our proposed implementation with the one contained in [17].

The aim of the work presented in this paper is to propose a novel reduced architecture of the SpaceFibre protocol (from now R-SpaceFibre) that optimises resources utilisation at the expenses of reduced features. Furthermore, the proposed modifications to the standard are intended to be compatible with a fully compliant implementation of it. R-SpaceFibre has been realised starting from the IngeniArs SpaceFibre CODEC IP. Since the aim of this paper is to present a modified and reduced version of the SpaceFibre standard, the results presented are all referred to the same full SpaceFibre CODEC (from IngeniArs). The obtained resource reduction is then theoretically achievable also on other different implementations.

In Section 2, architectural design choices for the development of R-SpaceFibre are presented focusing on the different protocol layers. In Section 3, the hardware set-up chosen to validate R-SpaceFibre and to demonstrate interoperability with a SpaceFibre CODEC is described and resources utilisation results on different space-grade FPGAs are presented. In Section 4, a possible use case of the R-SpaceFibre IP core is discussed. Finally, Section 5 summarises the presented work and its outcomes.

2. R-SpaceFibre Design: Features and Trade-Offs

To understand the design choices made in the R-SpaceFibre development, it is necessary to briefly introduce the SpaceFibre protocol stack.

The stack is composed of five layers, as shown in Figure 1:

- The **Network layer** is responsible for transferring both data and broadcast messages through the SpaceFibre network.
- The **Data Link layer** is responsible for the FDIR system that implements a retry mechanism to resend corrupted frames. Moreover, it handles independent flow of information through Virtual Channel Buffers (VCBs), and it manages the QoS, broadcast service and data scrambling.
- The **Multi lane layer** is responsible for synchronising a link composed by more than 1 lane (up to 16). This is an optional layer.
- The **Lane layer** is responsible for establishing the communication between two ends.
- The **Physical layer** is responsible for serialising, transmitting, receiving and de-serialising data.

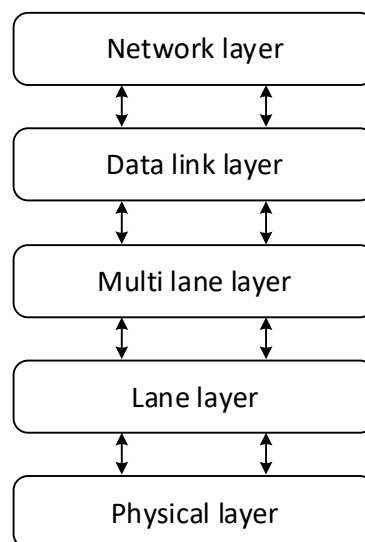


Figure 1. SpaceFibre protocol stack.

The Network layer has not been included in R-SpaceFibre design as it is realised as an optional upper layer, and it is not needed for point-to-point links. R-SpaceFibre indeed has been designed for satellites with stringent requirements in terms of size and volume. These satellites often mount a reduced number of devices and usually a single payload, thus point-to-point high-speed connections are definitively preferred against more complex network topologies. The Multi lane layer is not included as well: it allows increasing the maximum achievable data-rate and system reliability adding redundancy but requiring considerably more hardware resources [18].

For this reason, the reference SpaceFibre IP cell of this study is composed only by the core layers: Data link, Lane and Physical Layer. Most of the architectural modifications aim at reducing the hardware resources in the Data link layer, which is the most complex layer in terms of area occupation [19]. Some of its features are not essential to the core communication process, thus are removable from the design, accepting some performance degradation. Indeed, the Lane layer, which is responsible for initialising the link, and the Physical layer are fully compliant with the standard, as no significant optimisations are possible.

Schematic descriptions of both SpaceFibre and R-SpaceFibre Data link layers are presented in Sections 2.1 and 2.2, in order to point out architectural innovations and shared features. The internal architectural description of each building block is not described as the aim of this work is to propose a high level architecture which saves resources employing a subset of the building block already present

in a regular SpaceFibre interface. For a full description of the layer, please refer to the SpaceFibre standard [2].

2.1. SpaceFibre Data Link Layer

The Data link layer of a fully compliant version of the protocol is described in this paragraph, with reference to a simplified architectural block diagram shown in Figure 2. It is composed by a Medium Access Controller (MAC), a transmission block (*Tx Block*) and a receiving block (*Rx Block*).

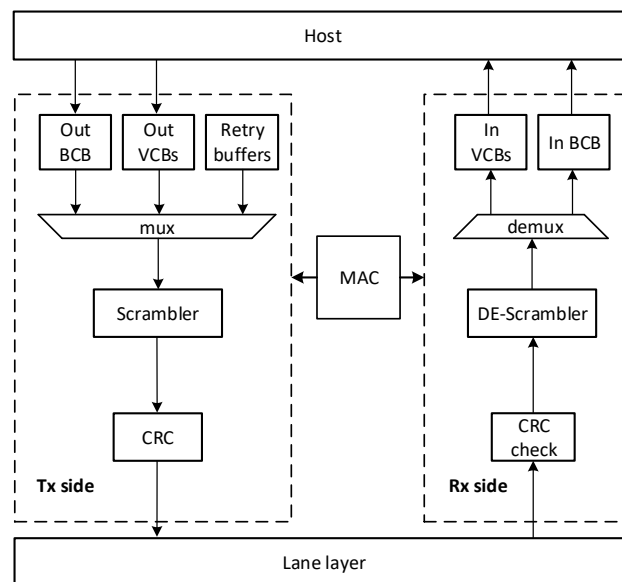


Figure 2. SpaceFibre data link layer block diagram.

The data interface with the host is realised through VCs, both on the RX side (*Out VCBs*) and on the TX side (*In VCBs*). They allow handling independent flows of information across a single physical link. There are up to 32 different VCs available. A special interface is dedicated to broadcast (BC) messages, with one Rx buffer, Out Broadcast Channel Buffer (*Out BCB*), and one TX buffer, Input Broadcast Channel Buffer (*In BCB*).

The MAC is the core of the Data link layer: it manages data framing and it is responsible for synchronising the operations between transmitting and receiving side. Moreover, the MAC implements the QoS: it continuously schedules which VC shall send data based on different parameters such as priority and reserved bandwidth.

The Data link layer also implements retry buffers to resend data in the case the far-end of the link detects a communication error and asks for retransmission. The Retry Buffers block contains three buffers: the Data Retry Buffer, the Flow Control Token (FCT) Retry Buffer and the Broadcast Retry Buffer.

Data or BC are read out from the selected VC or from the corresponding Retry Buffer and framed appropriately. Then, data frame could be scrambled to reduce electromagnetic emissions and eventually the Cyclic Redundancy Check (CRC) [20] block computes a numeric field to be added at the end of each data frame, broadcast message and control words that require it. The far-end will re-compute independently this value and will compare it with the received one, in order to detect accidental errors occurred during the communication process and promptly ask for retransmission of corrupted frames or control words.

Conversely, on the Rx side, received data frames, BC messages and control words are handled. First, CRC Check block controls the correctness of the CRC value embedded in received data. In the case of errors, a retry request is sent to the far-end. If no errors are detected, data frame may be de-scrambled and stored in the appropriate *In VCB*, whereas broadcast messages are stored in the *In BCB*. Control

words are consumed for managing the communication process. For detailed information, please refer to the SpaceFibre standard [2].

2.2. R-SpaceFibre Data Link Layer

This section describes the proposed architecture of R-SpaceFibre Data link layer. A simplified architectural block diagram of the layer is shown in Figure 3. R-SpaceFibre major innovation is the exclusion of the retry mechanism from the protocol stack. The first consequence is that none of the Data Retry Buffer, the Broadcast Retry Buffer or the FCT Retry Buffer has to be included in the R-SpaceFibre Data link layer, with consistent savings of logical resources. This comes at the price of no data-retry if an error occurs with consequent data loss. However, the system is designed to continue the communication process, and a corrupted datum is passed to the higher level of the protocol. In fact, target applications such as SAR imagers can withstand the corruption of a single bit or the loss of a data word. The SpaceFibre CODEC properly works with a maximum Bit Error Rate (BER) of 10^{-5} [11]. However, typical BERs for satellite applications are in the order of 10^{-10} , which is much lower than the worst case handled by the SpaceFibre protocol. The application system engineer will decide whether complexity reduction at the price of data retry removal is acceptable for their application, depending on link criticality, orbit (BER), etc.

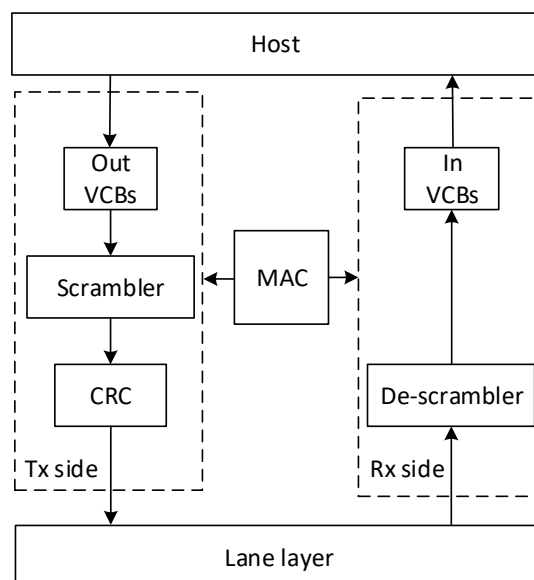


Figure 3. R-SpaceFibre data link layer block diagram.

R-SpaceFibre is designed mainly for point-to-point communication. For this reason, the broadcast service, responsible for sending short messages to all the nodes of a network, is unnecessary. Thus, *In* and *Out BCBs* have been excluded from the architecture. The removal of retry and broadcast services implies that also all QoS mechanisms (i.e., VCs scheduling and bandwidth reservation) become less complex, thus the MAC can be simplified.

The CRC Check block has been removed from the Rx Block: consequently, the IP core is no more able to signal to the far-end if received data frames or control words are corrupted. The near-end needs to acknowledge also corrupted data frames, so that compatibility with a full implementation of the standard at the far-end is maintained. More details about R-SpaceFibre compatibility with a full SpaceFibre interface are provided in Section 2.3. On the contrary, the CRC on the Tx Block cannot be removed: it has to be maintained in order to be compatible with the SpaceFibre protocol. Indeed, a standard SpaceFibre endpoint always performs CRC check on received data frames and control words; without it, the CODEC would automatically consider all received data as corrupted. Nevertheless, the

CRC block could be excluded from the Tx Block if the R-SpaceFibre just needs to communicate with R-SpaceFibre nodes. The proposed design includes optional data scrambling on both Tx and Rx side.

The SpaceFibre standard establishes that the number of VCs shall to be between 1 and 32. R-SpaceFibre can handle an arbitrary number of VCs, but, to reduce the impact on resource usage, the number of VCs shall be limited to one or two. In particular, by choosing to use only one VC the QoS is drastically reduced, together with the multiplexing logic between VCs and the lower part of the Data link layer. Both standard and literature [17] suggest that the QoS could be completely eliminated in an implementation of just of one or two VCs, but this implies that babbling node protection is not provided [2].

Finally, VCs dimension is set to be 256 data words, which is the minimum dimension according to the standard. In conclusion, the most relevant design changes in the R-SpaceFibre development are the removal of:

- Retry buffers, from the Tx Block
- Broadcast service, from the entire Data link Layer
- CRC check block, from the Rx Block
- Several QoS mechanism, from the MAC

2.3. Fault Tolerance and Compatibility with Full SpaceFibre Interfaces

R-SpaceFibre is meant to be used as lightweight high performance high speed link. It is a slightly modified version of the SpaceFibre protocol, fully interoperable with a SpaceFibre endpoint but compliant just to a subset of the SpaceFibre standard requirements, in order to reduce system complexity. When two Reduced SpaceFibre (R-SpFi) endpoints communicate together, no restrictions on protocol operations arise, also in the case of Single Event Upset (SEU) bit-flip errors happening on the link. The received corrupted packet will be handled at higher levels of the protocol, application dependent. However, it is possible to connect R-SpaceFibre and SpaceFibre endpoints. Coherent data transmission and reception is guaranteed, discarding received broadcast messages and ignoring retry requests. R-SpaceFibre is programmed to discard any retry request from a far-end. Consequently, if an error occurs during the transmission of a packet from full SpFi interface to a reduced SpFi endpoint, no retry request is given and the received corrupted packet is transmitted at the higher level of the protocol, where it will be handled according to the application requirements. Indeed, R-SpaceFibre has been designed for communication link involving payloads streaming large amount of data that do not contain safety-critical information. On the other hand, when a SEU occurs in the transmission of a packet from a R-SpFi interface to a full SpaceFibre interface, the second one will constantly try to ask for data retransmission, blocking the traffic on that Virtual Channel. This can be easily handled programming a reset interface to be given in order to re-establish the communication flow.

3. Hardware Implementation

3.1. Hardware Setup and Tests

A R-SpaceFibre implementation has been tested and validated on a Xilinx ZC706 evaluation board [21]. The System on a Chip (SoC) mounted embeds one Zynq-7000 XC7Z045-2FFG900C FPGA, one Cortex A9 processor and up to 16 GTX transceivers. The hardware setup is shown in Figure 4: the programmable logic of the FPGA is used to implement two SpFi CODECs, which communicate using two GTXs. The transceivers are externally connected through a Xilinx FMC XM104 [22] expansion board, using e-SATA cables. The Cortex A9 generates and consumes packets to be transmitted or received over the SpaceFibre link, as described in [18]. The system automatically checks whether all received packets are correct.

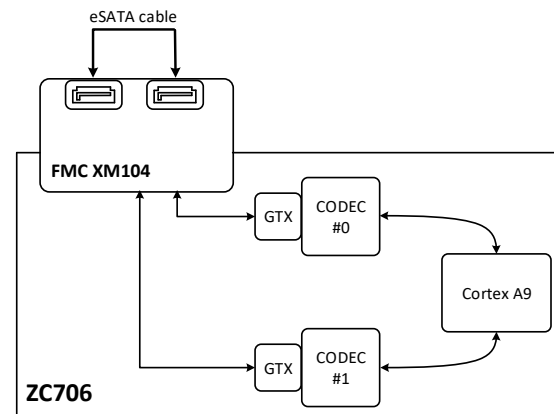


Figure 4. Architectural diagram of testbed.

To validate R-SpaceFibre, two configurations were tested:

- CODEC 0 is a R-SpaceFibre CODEC and CODEC 1 is a SpaceFibre CODEC (CONFIG1);
- CODEC 0 and CODEC 1 are both R-SpaceFibre CODECs (CONFIG2).

The system was intensively stimulated through an appropriate hardware validation plan, focusing on Data Link layer functionalities. In particular, QoS features, VCs scheduling and configurations with and without data scrambling were tested. Furthermore, errors (i.e., bit flips) were injected into the data stream in order to verify the behaviour of the CODECs both in CONFIG1 and CONFIG2.

In CONFIG1 tests, a deadlock arises when the SpaceFibre node send too many retry request to the R-SpaceFibre node, as described in Section 2.2. In particular, tests show that the time between two consecutive deadlocks increase decreasing packet dimension. In fact, the FCT counter is incremented of a number equal to the number of words composing a correctly received packet. CONFIG2 tests show data packet loss as expected, but no deadlock situation arises because R-SpaceFibre considers valid all received packets. The performed tests validated R-SpaceFibre: the reduced CODEC demonstrated to be able to communicate with a version of the CODEC compliant with the SpaceFibre standard, as well with another reduced implementation.

3.2. Resources Utilisation and Power Consumption

In this section, resource utilisation after carrying out logic synthesis, implementation and place and route on different space-grade FPGAs are presented. In particular, the number of Look-Up-Tables (LUTs), D-edge triggered flip-flops (DFF) and Random Access Memory (RAM) blocks necessary to implement one R-SpaceFibre CODEC are reported.

Table 1 shows the resource utilisation for the R-SpaceFibre CODEC on a Xilinx Zynq-7000, on a Xilinx Virtex 5, on a Microsemi RTG4 and on a Microsemi RTAX2000. These numbers refer to a R-SpaceFibre implementation with a single 256 words VCB and 62.5 MHz clock frequency, achieving a data-rate of 2.5 Gbps (SpaceFibre, and consequently R-SpaceFibre, shall transmit one 40 bit word per clock cycle).

Table 1. R-SpaceFibre COder-DECoder (CODEC) resource utilisation on different Field Programmable Gate Arrays (FPGAs).

FPGA	VC	LUT	util% LUT	DFF	util% DFF	BlockRAM-FIFO
Zynq-7000	1	1675	0.73%	1791	0.4%	1.5
	2	2248	1%	2345	0.5%	2
Virtex 5	1	1682	2.05%	1103	1.35%	2
	2	2662	3.25%	1653	2.02%	4
Microsemi RTG4	1	2561	2.07%	1227	1.19%	16
	2	4122	3.47%	1892	2.00%	32
Microsemi RTAX2000	1	3479	16.18 %	1530	14.23%	4
	2	5356	24.91%	2641	24.56%	8

Finally, Table 2 shows a comparison between hardware resources of R-SpaceFibre and IngeniArs SpaceFibre CODEC IP on a Microsemi RTAX2000 and on a Microsemi RTG4. The CODECs implemented still has a single 256 words VCB, and the target frequency is 62.5 MHz for RTG4 and only 40 MHz for the RTAX2000, due to the lower performances of the FPGA, with a maximum data-rate of 1.6 GHz. This is acceptable as the aim of this paper is to demonstrate the possibility to reduce hardware resources needed by a SpaceFibre end-node. It is clearly observable that R-SpaceFibre considerably reduces (more than 40%) the hardware resources needed for its implementation on both RTAX2000 and RTG4. In particular, RTAX2000 has limited number of hardware resources and, if compared with RTG4 and R-SpaceFibre implementation, allows saving a consistent part of synthesisable logic.

Table 2. SpaceFibre vs. R-SpaceFibre resources utilisation on Microsemi RTAX2000 and RTG4.

CODEC	RTAX2000		RTG4		Zynq-7000		Virtex 5	
	LUT	util%	LUT	util%	LUT	util%	LUT	util%
SpaceFibre	5830	27.11%	4632	3.98%	2800	1.28%	2919	3.56%
R-SpaceFibre	3479	16.18%	2561	2.07%	1675	0.73%	1682	2.05%
% Reduction	2351	40.33%	2071	44.71%	1125	40.18%	1237	42.39%
	DFF	util%	DFF	util%	DFF	util%	DFF	util%
SpaceFibre	2605	24.23%	1994	2.24%	2441	0.56%	1702	2.08%
R-SpaceFibre	1530	14.23%	1227	1.19%	1791	0.4%	1103	1.35%
% Reduction	1075	41.27%	767	38.46%	650	26.63%	599	35.20%
	RAM	util%	RAM	util%	RAM	util%	RAM	util%
SpaceFibre	12	18.75%	39	1.10%	5.5	1.28%	7	2.35%
R-SpaceFibre	4	6.25%	16	0.45%	1.5	0.28%	2	0.67%
% Reduction	8	66.66%	23	58.97%	4	72.72%	5	71.42%

Post-layout simulations were carried out on a Xilinx Zynq-7000 to estimate the power consumption of both R-SpaceFibre and SpaceFibre IP cores. The results were obtained with 200 μ s simulations, corresponding to the transmission of 520 kBit of data (divided in 36 bit data words). The CODECS were implemented with one VC and the operating frequency was set at 62.5 MHz. The results are shown in Table 3.

Table 3. R-SpaceFibre and SpaceFibre power consumption.

CODEC	Dynamic Power (mW)	Static Power (mW)	Total Power (mW)
R-SpaceFibre	49	199	248
SpaceFibre	55	199	254

Power consumption results similar for the two implementations. In fact, R-SpaceFibre does not implement broadcast service and the retry mechanism. However, in a standard SpaceFibre

communication link, those mechanism are rarely activated. Thus, the small difference between the two power consumption can be attributed to the fact that SpaceFibre CODEC computes the CRC code on received data frames.

4. R-SpaceFibre Use Case Scenario: CubeSats

R-SpaceFibre combines high data-rate capability with small hardware resources occupation, accepting reduced performances in terms of error recovery. This innovative technology is a suitable solution for applications with relaxed requirements in terms of bit error rate and stringent requirement of size, volume and cost such as CubeSats. Indeed, this class of small spacecraft needs optimised on-board components. In this section, CubeSats features and applications are analysed pointing out why R-SpaceFibre represents a valuable solution for enhancing CubeSats data-rate capability at low cost of volume and price.

CubeSat is a class of nanosatellite (with a mass between 1 and 10 Kg) standardised by the California Polytechnic State University in 1999. They are made up of $10\text{ cm} \times 10\text{ cm} \times 10\text{ cm}$ units (1U) with a maximum weight of 1.33 Kg [16] that could be assembled to compose larger satellites (i.e., 2U, 3U, 6U and 12U).

In the last few years, CubeSats have become more and more popular and the number of CubeSats launched into space per year is growing, as shown in Figure 5. According to the CubeSat database of the St. Louis University [23], 796 CubeSats had been launched as of 31 December 2018.

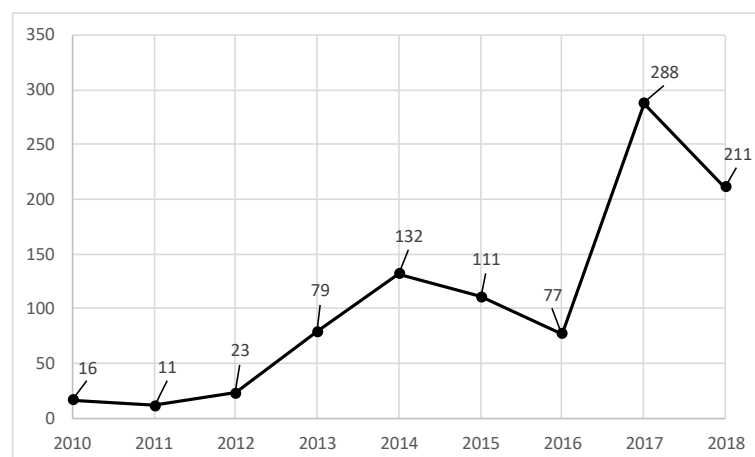


Figure 5. Number of CubeSats launched per year.

CubeSats, initially developed for educational purposes, despite their limitation in terms of mass and volume are emerging as important technological platforms, especially for Earth observation application, and represent a cost-effective and fast-to-launch solution [24].

A possible architecture for CubeSat on-board electronics is shown in Figure 6. The Control Unit manages and configures other units through command and control interfaces, characterised by low bandwidth and strict reliability constraints. The payload is connected through a high-speed link to an Instrumentation Control Unit that elaborates received data. Data are stored in a Mass Memory, which is connected as well through a high-speed link, waiting to be sent back to earth. Considering their limitation in terms of mass and volume, CubeSats cannot mount a great number of instruments and the data-handling subsystem is generally as simple as possible privileging point-to-point connections to more complex network topologies.

R-SpaceFibre may be a feasible solution for high-speed serial links in CubeSats data-handling subsystem, combining SpaceFibre cable mass reduction and performances with a small footprint on FPGA devices, which allows system designer to potentially fit it in FPGAs which share other tasks (i.e., the Instrument Control Unit, ICU). According to Table 1, it is possible to implement two

R-SpaceFibre interfaces (i.e., necessary to the ICU of Figure 6) on a small footprint FPGA like the RTAX2000, employing only 32.36% of the total LUT, 28.46% of DFF and 12.50% of RAM with large part of the FPGA still available for other mission dependent features implementation. The same architecture implementing a regular SpaceFibre needs 54.22% of LUT, 48.46% of DFF and 37.5% of RAM, offering less possibility to integrate extra features on the same device. R-SpaceFibre could help to reduce the number of devices embedding more features on the same FPGA. It also reduces the overall electronic price, which represent a key advantage for low cost missions, such as CubeSats.

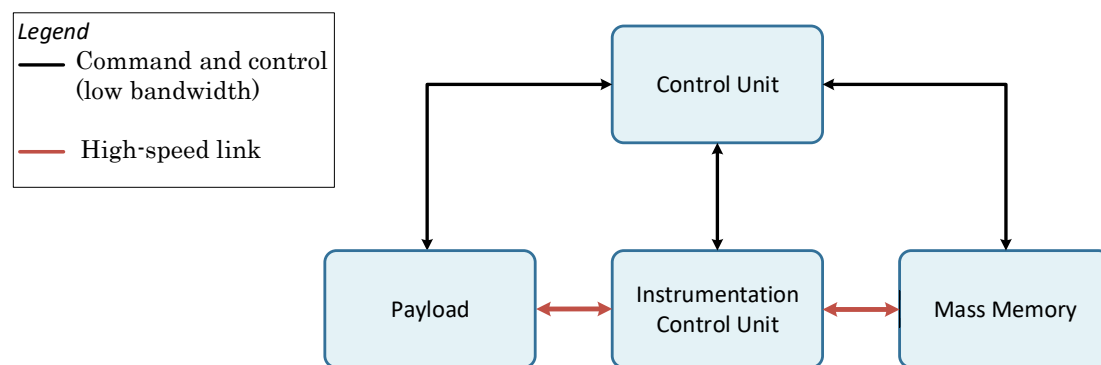


Figure 6. CubeSat on-board electronics architecture.

The state-of-the-art for rad-hardened CubeSat data-handling subsystem is the Sphinx Deep Space Command and Data Handling (CDH) board [25] developed by the Jet Propulsion Laboratory (JPL) that will be mounted on the Lunar Flashlight [26] and NEAScout missions [27]. The Sphinx CDH has a mass of less than 200 g and may be housed in a 10 cm × 10 cm area, with a maximum power consumption of 7 W. Considering the communication interfaces, the Sphinx mounts four SpaceWire ports that have a maximum data-rate of 400 Mbps and seems not to be the ideal solution for increasing CubeSat data-rate capability: four separate interfaces require separate connectors and cables, which could represent an issue in volume contained systems such as CubeSats. Moreover, the overall data-rate is still below the one easily obtainable with one R-SpaceFibre interface. The Sphinx CDH interface and R-Spacefibre may be compared considering hardware resources needed to be implemented on a FPGA device. Table 4 shows the results obtained for R-SpaceFibre and four SpaceWire interfaces realised, respectively, by IngeniArs [28] and STAR-Dundee [29] on a RTAX2000. The two communication interfaces need similar hardware resources, but R-SpaceFibre increases the achievable data-rate from 1.6 Gbps of the Sphinx SpaceWire solution to 1.8 Gbps and can be further improved to 3.125 Gbps implementing the design on higher performance FPGAs such as RTG4. Furthermore, R-SpaceFibre reduces cable volume and mass: considering state-of-the-art cables from Axon, the Sphinx requires four SpaceWire cables (one per port), whose weight is 42 g/m [30] for a total of 168 g/m. On the other hand, R-SpaceFibre requires only one SpaceFibre cable, weighting 15 g/m [31], achieving around 90% mass reduction, as shown in Table 5.

Table 4. R-SpaceFibre and four SpaceWire interfaces resources utilisation on RTAX2000.

Resources	R-SpaceFibre	4 SpW (IngeniArs)	4 SpW (STAR-Dundee)
LUT	16.18%	11.88%	7.16%
DFF	14.23%	14.12%	17.22%
Block RAM	4	8	8

Table 5. R-SpaceFibre interface vs. four SpaceWire interfaces cable mass.

Solution	Number of Cables	Mass (g/m)
R-SpaceFibre	1	15
4 SpaceWire	4	168

Currently, CubeSat missions already plan to mount SAR and Interferometric SAR (InSAR) as payloads. Those instruments require data-rate in the order of Gbps, which is not easily targetable with state-of-the-art technology. For example, SRI International has prototyped the CubeSat Imaging Radar for Earth Science (CIRES), designed for a 6U CubeSat constellation [32] and capable of a moderate resolution (25 m). Moreover, other present and future missions, such as Capella 1 [33] and Iceye [34], will mount SAR as payloads, while Intuition-1 [35], HyperCube [36] and Waypoint 1 [37] will mount hyper-spectral imagers. On-board data-handling bandwidth of these missions is expected to be in the order of Gbps, falling in the operational range of R-SpaceFibre.

This brief analysis suggests that R-SpaceFibre may be a valid solution for on-board data handling of CubeSats mounting high resolution payloads in the near future: even if R-SpaceFibre performances are reduced compared to full SpaceFibre implementations, it allows significantly reducing the required hardware resources, combining high data-rate with other advantages proper for the SpaceFibre protocol.

5. Conclusions

R-SpaceFibre, a modified implementation of the SpaceFibre standard with smaller footprint and limited features, is presented in this paper. A brief analysis of satellite on-board high-speed serial links state-of-the-art is given, pointing out the reasons that lead ESA to support the development of SpaceFibre protocol. The idea of having a high-speed protocol with lower resource utilisation is exploited in some application scenarios, proposing R-SpaceFibre as a suitable solution. Starting from SpaceFibre standard analysis, the design choices behind R-SpaceFibre are described, showing and analysing design trade-off functionalities versus complexity. Resource utilisation results on space-grade devices are presented and compared with the one of a complete SpaceFibre implementation, showing that a resource reduction above 40% is achieved. A hardware validation set-up is then presented and two different testing configurations are described, demonstrating R-SpaceFibre compatibility with a complete version of the CODEC. Finally, CubeSat use case is analysed, highlighting how R-SpaceFibre represents an enabling technology for small satellite high-speed communication links, increasing data-rate and limiting the overall mass and volume of the electronic system.

Author Contributions: The research and the article preparation were carried out by G.D. and P.N. under the supervision of L.F. and D.D.

Funding: IngeniArs SpaceFibre technologies have been developed in the framework of the SIMPLE project (Spacefibre IMPLementation design & test Equipment). This project received funding from the European Unions Horizon 2020 research and innovation programme under Grant Agreement No. 757038.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

BC	broadcast
CDH	Command and Data handling
CIRES	CubeSat Imaging Radar for Earth Science
COTS	Commercial-Off-The-Shelf
CRC	Cyclic Redundancy Check
ECSS	European Cooperation for Space Standardization
EGSE	Electrical Ground Segment Equipments
ESA	European Space Agency
FCT	Flow Control Token
FDIR	Fault Detection Isolation and Recovery
FPGA	Field Programmable Gate Array
In BCB	In Broadcast Channel Buffer
InSAR	Interferometric SAR
IP	Intellectual Property
JAXA	Japan Aerospace eXploration Agency
JPL	Jet Propulsion Laboratory
LEO	Low Earth Orbit
LUT	Look-Up-Tables
MAC	Medium Access Controller
NASA	National Aeronautics and Space Administration
Out BCB	Out Broadcast Channel Buffer
QoS	Quality of Service
Reg	register
SAR	Synthetic Aperture Radars
VC	Virtual Channel
VCB	Virtual Channel Buffers

References

1. Parkes, S.; McClements, C.; McLaren, D.; Florit, A.F.; Villafranca, A.G. SpaceFibre: A multi-Gigabit/s interconnect for spacecraft onboard data handling. In Proceedings of the 2015 IEEE Aerospace Conference, Big Sky, MT, USA, 7–14 March 2015.
2. SpaceFibre standard, ECSS-E-ST-50-11C-DIR1. Available online: <http://ecss.nl/standard/ecss-e-st-50-11c-dir1/> (accessed on 5 May 2019).
3. Parkes, S.; Armbruster, P. SpaceWire A Spacecraft Onboard Network for Real-Time Communications. In Proceedings of the 14th IEEE-NPSS Real Time Conference, Stockholm, Sweden, 4–10 June 2005.
4. Saponara, S.; Fanucci, L.; Tonarelli, M.; Petri, E. Radiation Tolerant SpaceWire Router for Satellite On-Board Networking. *IEEE Aerosp. Electron. Syst. Mag.* **2017**, *22*, 3–12. [[CrossRef](#)]
5. Mason, A.; Parkes, S. Using SpaceWire with LabVIEW. In Proceedings of the 2014 International SpaceWire Conference, Athens, Greece, 22–26 September 2014.
6. Rivera, J.P.; Sabater, N.; Tenjo, C.; Vicen, J.; Alonso, L.; Moreno, J. Synthetic scene simulator for hyperspectral spaceborne passive optical sensors. Application to ESA's FLEX/sentinel-3 tandem mission. In Proceedings of the 2014 6th Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS), Lausanne, Switzerland, 24–27 June 2014.
7. Toan, T.L.; Chave, J.; Dall, J.; Papathanassiou, K.; Paillou, P.; Rechstein, M.; Quegan, S.; Saatchi, S.; Seipel, K.; Shugart, H.; et al. The Biomass Mission: Objectives and Requirements. In Proceedings of the 2018 IEEE International Geoscience and Remote Sensing Symposium, Valencia, Spain, 22–27 July 2018.
8. Koga, R.; Yu, P.; George, J. Single Event Effects and Total Dose Test Results for TI TLK2711 Transceiver. In Proceedings of the 2008 IEEE Radiation Effects Data Workshop, Tucson, AZ, USA, 14–18 July 2008, pp. 69–75.
9. Fuller, S.; Cox, T. Anatomy of a Forward-Looking Open Standard. *Computer* **2002**, *35*, 140–141. [[CrossRef](#)]

10. Kopetz, H.; Ademaj, A.; Grillinger, P.; Steinhammer, K. The Time-Triggered Ethernet (TTE) Design. In Proceedings of the 8th IEEE International Symposium on Object-Oriented Real-Time Distributed Computing, Auckland, New Zealand, 13–17 April 2015.
11. Davalle, D.; Leoni, A.; Dello Sterpaio, L.; Fanucci, L. Design and implementation of test equipment for SpaceFibre links: SpaceFibre. In Proceedings of the 7th International SpaceWire Conference, Yokohama, Japan, 25–27 October 2016, pp. 1–5.
12. Siegle, F.; Habinc, S.; Both, J. SpaceFibre Port IP Core (GRSPFI): SpaceFibre, poster paper. In Proceedings of the 7th International SpaceWire Conference, Yokohama, Japan, 25–27 October 2016.
13. Leoni, A.; Nannipieri, P.; Fanucci, L. VHDL Design of a SpaceFibre Routing Switch. *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.* **2019**, E102-A, 729–731. [[CrossRef](#)]
14. Parkes, S.; Florit, A.F.; Villafranca, A.G.; McClements, C.; McLaren, D. SpaceFibre network and routing switch. In Proceedings of the 2017 IEEE Aerospace Conference, Big Sky, MT, USA, 4–11 March 2017, pp. 1–7.
15. Leoni, A.; Nannipieri, P.; Davalle, D.; Fanucci, L.; Jameux, D. SHINe: Simulator for Satellite on-Board High-Speed Networks Featuring SpaceFibre and SpaceWire Protocols. *Aerospace* **2019**, 6, 43. [[CrossRef](#)]
16. CubeSat design specification Rev. 12, 2009. Available online: <http://www.cubesat.org/resources/> (accessed on 5 May 2019).
17. Rowlings, M.; Suess, M. An Experimental Evaluation of SpaceFibre Resource Requirements. In Proceedings of the 2014 International SpaceWire Conference, Athens, Greece, 22–26 September 2014.
18. Nannipieri, P.; Dinelli, G.; Davalle, D.; Fanucci, L. A SpaceFibre multi lane codec System on a Chip: Enabling technology for low cost satellite EGSE. In Proceedings of the PRIME 2018—14th Conference on Ph.D. Research in Microelectronics and Electronics, Prague, Czech Republic, 2–5 July 2018; 8430317, pp. 173–176.
19. Parkes, S.; McClements, C.; McLaren, D.; Youssef, B.; Ali, M.S.; Florit, A.F.; Villafranca, A.G. SpaceWire and SpaceFibre on the Microsemi RTG4 FPGA. In Proceedings of the 2016 IEEE Aerospace Conference, Big Sky, MT, USA, 5–12 March 2016, pp. 1–8.
20. Koopman, P.; Chakravarty, T. Cyclic Redundancy Code (CRC) Polynomial Selection for Embedded Networks. In Proceedings of the International Conference on Dependable Systems and Networks, Florence, Italy, 28 June–1 July 2004, pp. 145–154.
21. ZC706 Evaluation Board for the Zynq-7000 XC7Z045 All Programmable SoC User Guide. Available online: <https://www.xilinx.com/support/documentation/boardsandkits/zc706/ug954-zc706-eval-board-xc7z045-ap-soc.pdf> (accessed on 5 May 2019).
22. FM-XM104 Connectivity Card User Guide. Available online: <https://www.xilinx.com/support/documentation/boardsandkits/ug536.pdf> (accessed on 5 May 2019).
23. St. Louis University CubeSat database. Available online: <https://sites.google.com/a/slu.edu/swartwout/home/cubesat-database#plots> (accessed on 5 May 2019).
24. Poghosyan, A.; Golkar, A. CubeSat evolution: Analyzing CubeSat capabilities for conducting science missions. *Prog. Aerosp. Sci.* **2017**, 88, 59–83. [[CrossRef](#)]
25. Kahn, P.; Imken, T.; Elliott, J.; Sherwood, B.; Frick, A.; Sheldon, D.; Lunine, J. Environmental design implications for two deep space SmallSats. *Acta Astronaut.* **2017**, 139, 390–395. [[CrossRef](#)]
26. Cohen, B.A.; Hayne, P.O.; Greenhagen, B.T.; Paige, D.A. Lunar Flashlight: Exploration and science at the moon with a 6U CubeSat. In Proceedings of the 2015 Annual Meeting of the Lunar Exploration Analysis Group, Columbia, MD, USA, 20–22 October 2015.
27. McNutt, L.; Johnson, L.; Clardy, D.; Castillo-Rogez, J.; Frick, A.; Jones, L. Near-Earth Asteroid Scout. In Proceedings of the AIAA Space 2014 Conference and Exposition, San Diego, CA, USA, 4–7 August 2014.
28. IngeniArs SpaceWire. Available online: <https://www.ingeniars.com/> (accessed on 5 May 2019).
29. StarDundee SpaceWire. Available online: <https://www.star-dundee.com/> (accessed on 5 May 2019).
30. SpaceWire cable datasheet. Available online: <http://www.axon-cable.com/publications/HIGH-SPEED-LINKS.pdf> (accessed on 5 May 2019).
31. SpaceFibre cable datasheet. Available online: http://www.axon-cable.com/publications/High_Speed_Links-SPACEFIBRE.pdf (accessed on 5 May 2019).
32. Wye, L.; Lee, S. SRI CubeSat imaging radar for earth science (SRI-CIRES). In Proceedings of the 2016 Earth Science Technology Forum, Annapolis, MD, USA, 14–16 June 2016.
33. Capella CubeSat. Available online: <https://www.capellaspace.com/> (accessed on 5 May 2019).
34. Iceye CubeSat. Available online: <https://www.iceye.com/resources/technology/> (accessed on 5 May 2019).

35. Intuition CubeSat. Available online: <https://www.fpspace.com/en> (accessed on 5 May 2019).
36. Hypercube CubeSat. Available online: https://www.harris.com/sites/default/files/downloads/solutions/55493_hypercube_data_sheet_v2_2_final_.pdf (accessed on 5 May 2019).
37. Waypoint CubeSat. Available online: <http://www.spacefab.us/space-telescopes.html> (accessed on 5 May 2019).



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).