SpaceFibre: Adaptive High-Speed Data-Link for Future Spacecraft Onboard Data Handling

Steve Parkes, Chris McClements, David McLaren,
Angel Monera Martinez,
Space Technology Centre, University of Dundee,
166 Nethergate, Dundee, DD1 4EE, UK
sparkes@computing.dundee.ac.uk

Albert Ferrer Florit, Alberto Gonzalez Villafranca,
STAR-Dundee Ltd.,
STAR House, 166 Nethergate, Dundee, DD1 4EE, UK

***Abstract* - SpaceFibre is a high-speed data-link technology being developed by the University of Dundee for ESA to support spacecraft onboard data-handling applications. SpaceFibre operates at 2.5 Gbits/s, can run over fibre optic or electrical media, provides galvanic isolation, includes Quality of Service (QoS) and Fault Detection Isolation and Recovery (FDIR) support, and provides low-latency signalling. It operates over distances of 5m with copper cable and 100 m or more with fibre optic cable. SpaceFibre supports multiple virtual channels running over a single physical link. QoS capabilities built into the SpaceFibre hardware allow the bandwidth and priority of each virtual channel to be specified. Traffic flow over each virtual channel then adapts automatically taking into account virtual channels that have data ready to send and available buffer space at the far end of the link, along with link bandwidth and priority allocation. The novel QoS mechanism is simple but powerful and also allows the automatic detection of "babbling idiots" and virtual channels that are sending much less data than expected. After a brief introduction the SpaceFibre QoS and FDIR capabilities are explained. The approach taken in validating the SpaceFibre protocols and current status of the SpaceFibre development activities are then described.**

***Keywords—**SpaceWire, SpaceFibre, networks, spacecraft onboard data-handling*

I. INTRODUCTION

There is a growing need for a standard high-speed serial data link for use onboard spacecraft to connect high data-rate instruments to the mass-memory unit, to interconnect memory modules within the mass-memory unit, and to connect the mass-memory unit to the downlink telemetry unit. SpaceFibre [1] [2] [3] is a very high-speed serial data-link being developed by the University of Dundee for ESA for use with high data-rate payloads. SpaceFibre is able to operate over fibre-optic and electrical cable and support data rates of 2 Gbit/s in the near future and up to 5 Gbit/s long-term. Multi-laning improves the data-rate further to over 20 Gbits/s. SpaceFibre complements the capabilities of the widely used SpaceWire onboard networking standard [4]. SpaceFibre provides 12 times the bandwidth of SpaceWire, and adds QoS and FDIR which are missing in SpaceWire. SpaceFibre also includes multi-lane capability with support for graceful degradation which is again missing in SpaceWire. SpaceFibre uses the same packet format as SpaceWire, making bridging between SpaceWire and SpaceFibre straightforward and allowing many SpaceWire links to be concentrated to run over a SpaceFibre link each occupying a separate virtual channel.

SpaceFibre provides a coherent quality of service mechanism able to support bandwidth reserved, scheduled and priority based qualities of service. It has integrated fault detection, isolation and recovery (FDIR) capabilities built into the hardware. Temporary faults on a link are recovered automatically without any loss of data. Permanent faults are reported rapidly or can be recovered with graceful degradation when using multiple-lanes.

II. SPACEFIBRE PROTOCOL STACK

The SpaceFibre protocol stack is illustrated in Fig. 1.

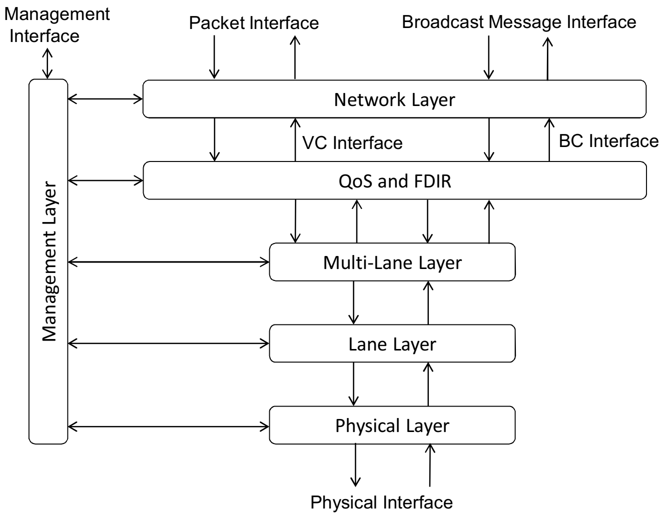


Fig. 1. SpaceFibre Protocol Stack

The network layer protocol is responsible for the transfer of application information over a SpaceFibre network. It provides two services: Packet Transfer Service and Broadcast Message Service. The Packet Transfer Service transfers SpaceFibre packets over the SpaceFibre network, using the same packet format and routing concepts as SpaceWire uses. SpaceFibre supports both path and logical addressing. The broadcast message service is responsible for broadcasting short messages (8 bytes) to all nodes on the network. These messages can carry time and synchronisation signals and be used to signal the occurrence of various events on the network.

The management layer is responsible for configuring, controlling and monitoring the status of all the layers in the SpaceFibre protocol stack. For example it can configure the QoS settings of the virtual channels in the QoS and FDIR layer.

The QoS and FDIR layer is responsible for providing quality of service and managing the flow of information over a SpaceFibre link. It frames the information to be sent over the link to support QoS and scrambles the packet data to reduce electromagnetic emissions. The QoS and FDIR layer also provides a retry capability, detecting any frames or control codes that go missing or arrive containing errors and resending them. With this inbuilt retry mechanism SpaceFibre is very resilient to transient errors.

The Multi-Lane layer is responsible for operating several SpaceFibre lanes in parallel to provide higher data throughput. In the event of a lane failing the Multi-Lane layer provides support for graceful degradation, automatically spreading the traffic over the remaining working links. When a lane stops working the remaining lanes are resynchronised and data sent across the available lanes.

The Lane layer is responsible for lane initialisation and error detection. In the event of an error the lane is automatically re-initialised. The Lane layer encodes data into symbols for transmission using 8B/10B encoding and decodes these symbols in the receiver. 8B/10B codes are DC balanced supporting AC coupling of SpaceFibre interfaces.

The Physical layer is responsible for serialising the 8B/10B symbols and for sending them over the physical medium. In the receiver the Physical layer recovers the clock and data from the serial bit stream, determines the symbol boundaries and recovers the 8B/10B symbols. Both electrical cables and fibre-optic cables are supported by SpaceFibre.

III. SpaceFibre Quality of Service

A SpaceFibre interface includes a number of virtual channels. Each provides a FIFO type interface similar to that of a SpaceWire link. When data from a SpaceWire packet is placed in a SpaceFibre virtual channel it is transferred over the SpaceFibre link and placed in the same numbered virtual channel at the other end of the link. Data from the several virtual channels are interleaved over the physical SpaceFibre connection. To support the interleaving, data is sent in short frames of up to 256 SpaceWire N-chars each. A virtual channel can be assigned a quality of service which determines the precedence with which that virtual channel will compete with other virtual channels for sending data over the SpaceFibre link. Priority, bandwidth reservation, and scheduled qualities of service can be supported all operating together using a simple precedence mechanism. The QoS mechanism adapts the traffic flow over the link to make maximum use of the link bandwidth while taking into account the priority, reserved bandwidth and flow-control for each virtual channel.

In this section the SpaceFibre quality of service mechanism is described.

A. Frames and Virtual Channels

To provide quality of service, it is necessary to be able to interleave different data flows over a data link or network. If a large packet is being sent with low priority and a higher priority one requests to be sent, it must be possible to suspend sending the low priority one and start sending the higher priority packet. To facilitate this SpaceWire packets are chopped up into smaller data units called frames. When the high priority packet requests to be sent, the current frame of the low priority packet is allowed to complete transmission, and then the frames of the high priority packet are sent. When all the frames of the high priority packet have been sent, the remaining frames of the low priority packet can be sent. Each frame has to be identified as belonging to a particular data flow so that the stream of packets can be reconstructed at the other end of the link.

Each independent data stream allowed to flow over a data link is referred to as a virtual channel (VC). Virtual channels are unidirectional and have a QoS attribute, e.g. priority. At each end of a virtual channel is a virtual channel buffer (VCB), which buffers the data. An output VCB takes data from the application and buffers it prior to sending it across the data link. An input VCB receives data from the data link and buffers it prior to passing it to the receiving application.

There can be several output virtual channels connected to a single data link, which compete for sending information over the link. A medium access controller determines which output virtual channel is allowed to send the next data frame. When an output VCB has a frame of data ready to send and the corresponding input VCB at the other end of the link has room for a full data frame, the output VCB requests the medium access controller to send a frame. The medium access controller arbitrates between all the output VCBs requesting to send a frame. It uses the QoS attribute of each of the requesting VCBs to determine which one will be allowed to send the next data frame.

B. Precedence

For the medium access controller to be able to compare QoS attributes from different output VCBs, it is essential that they are all using a common measure that can be compared. The name given to this measure is precedence. The competing output VCB with the highest precedence will be allowed to send the next frame.

C. Bandwidth Reservation

When connecting an instrument via a network to a mass memory, what the systems engineer needs to know is “how much bandwidth do I have to transfer data from the instrument to the mass memory?” Once the network bandwidth allocated to a particular instrument has been specified, it should not be possible for another instrument to intrude on the bandwidth allocated to that instrument. A priority mechanism is not suitable for this application. If an instrument with high priority has data to send it will hog the network until all its data has been sent. What is needed is a mechanism that allows bandwidth to be reserved for a particular instrument.

Bandwidth reservation calculates the bandwidth used by a particular virtual channel, and compares this to the bandwidth reserved for that virtual channel to calculate the precedence for that virtual channel. If the virtual channel has not used much reserved bandwidth recently, it will have a high precedence. When a data frame is sent by this virtual channel, its precedence will drop. Its precedence will increase again over a period of time. If a virtual channel has used more than its reserved bandwidth recently, it will have a low precedence.

A virtual channel specifies a portion of overall link bandwidth that it wishes to reserve and expects to use, i.e. its Expected Bandwidth. If a virtual channel has been allocated a portion of network bandwidth and it is not sending any data for a while, it will accumulate precedence so that when it has some data to send it will be more likely to be allowed to send that data. Precedence gradually increases for each virtual channel when it is not sending data. When a virtual channel sends a frame of data its precedence will drop an amount inversely proportional to its expected bandwidth. A virtual channel with 10% bandwidth allocation will drop twice as much as a virtual channel with 20% expected bandwidth. The precedence is updated every time a data frame for any virtual channel has been sent. To simplify the hardware required to calculate the precedence it is allowed to saturate at maximum and minimum values. This is illustrated in Fig. 2.

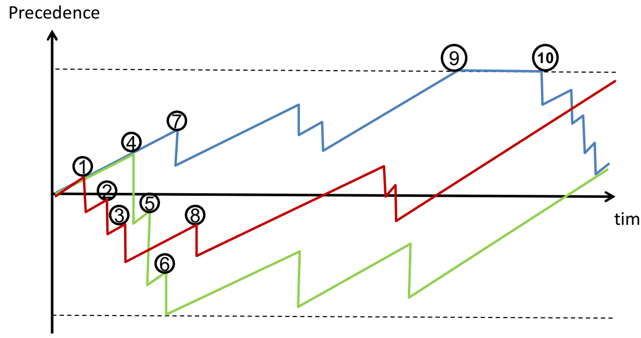


Fig. 2. Bandwidth Credit of Competing VCs

When a VC has a data frame ready to send and room for a full data frame at the other end of the link, it competes with any other VCs in a similar state, the one with the highest precedence being allowed to send the next data frame. At points (1), (2) and (3) the red VC has data to send and sends frames. At points (4), (5) and (6) the green VC has data to send and sends a data frame. At point (7) both the blue and the red VCs have data to send. The blue VC wins since it has the highest precedence. After this the red VC is allowed to send a further data frame at point (8).

At point (9) the precedence of the blue VC eventually saturates, reaching its maximum permitted value. Although more bandwidth should be accumulated after point (9) this is effectively ignored since the maximum possible bandwidth credit has been reached. At point (10) a frame is sent once more, resulting in a drop from the maximum bandwidth credit value.

The red and blue VCs have been allocated twice as much bandwidth as the green VC. The precedence of the green VC drops twice as far as the other VCs when it sends a frame.

If the precedence reaches the minimum possible value, it indicates that it is using more bandwidth than expected and a possible error may be flagged. This condition may be used to stop the VC sending any more data until it recovers some bandwidth credit, to help with "babbling idiot" protection where a device with a fault starts to send data unexpectedly.

Similarly if the precedence stays at the maximum possible precedence value for a relatively long period of time, the VC is using less bandwidth than expected and this condition can be flagged to indicate a possible error.

The component of precedence due to bandwidth reservation is called Bandwidth Credit.

D. Priority

The second type of QoS provided by VCs is priority. Each VC is assigned a priority value and the VC with the highest priority (lowest priority number) is allowed to send the next data frame as soon as it is ready. Fig. 3. shows three priority levels. SpaceFibre has 16 priority levels.

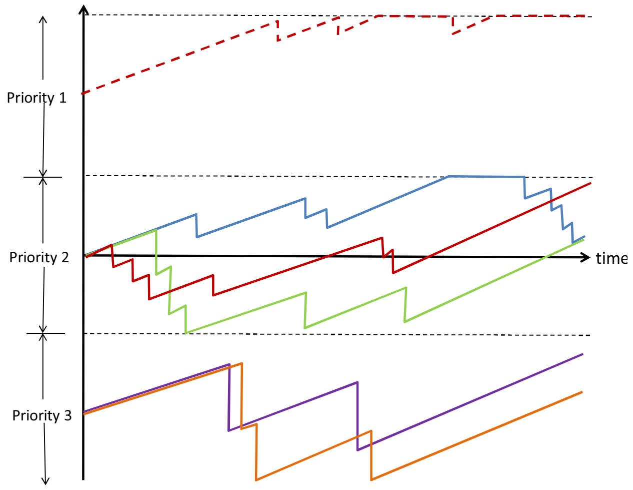


Fig. 3. Multi-Layered Precedence Priority QoS

Within any level there can be any number of VCs which compete amongst themselves based on their bandwidth credit. A higher priority VC will always have precedence over a lower priority VC unless its Bandwidth Credit has reached the minimum credit limit in which case it is no longer allowed to send any more data frames. This prevents a high priority VC from consuming all the link bandwidth if it fails and starts babbling. More than one VC can be set to the same priority level in which case those VC's will compete for medium access using bandwidth reservation.

E. Scheduled

To provide fully deterministic data delivery it is necessary for the QoS mechanism to ensure that data from specific virtual channels can be sent (and delivered) at particular times. This can be done by chopping time into a series of time-slots, during which a particular VC is permitted to send data frames. This is illustrated in Fig. 4.

Time-slot	1	2	3	4	5	6	7	8
VC 1	🟩							
VC 2		🟩				🟩		
VC 3				🟩				
VC 4								
VC 5							🟩	
VC 6			🟩					
VC 7								
VC 8			🟩					

Fig. 4. Scheduled Quality of Service

Each VC is allocated one or more time-slots in which it is permitted to send data frames. VC1 is scheduled to send in time-slot 1 and VC2 is scheduled to send in time-slots 2 and 6. The time-slot duration is a system level parameter, typically 100 µs, and there are 64 (TBC) time-slots.

During a time-slot, if the VC is scheduled to send in that time-slot, it will compete with other VCs also scheduled to send in that time-slot based on precedence (priority and bandwidth credit). A fully deterministic system would have one VC allowed to send in a time-slot.

The schedule is always operating. If a system engineer does not require the use of scheduling the schedule table is simply filled completely, allowing any VC to send in any time-slot, competing with precedence.

Scheduling can waste bandwidth if only one VC is allowed to send in a time-slot and that VC is not ready. To avoid this situation, the critical VC can be allocated a time-slot and given high priority. Another VC can be allocated the same time-slot with lower priority. In this way when that time-slot arrives the high priority VC will be allowed to send its data, but if it is not ready the VC with lower priority can send some data. This configuration is illustrated in Fig. 4. time-slot 3 and VCs 6 and 8.

Time-slots can be defined using broadcast messages to send start of time-slot signals or to send time information and having a local time counter which determines the start and end of each time-slot. A node designated as the network manager would send out these broadcast messages periodically. The SpaceFibre broadcast message mechanism supports both synchronisation and time distribution.

The SpaceFibre QoS mechanism is simple and efficient to implement and it provides bandwidth reservation, priority and scheduling integrated together, rather than as separate options. The data flow across the physical link adapts automatically to use the full link bandwidth while respecting the bandwidth allocation and priority of each virtual channel. Furthermore SpaceFibre QoS provides a means for detecting "babbling idiots" and for detecting nodes that have ceased sending data when they are expected to be sending information.

IV. SpaceFibre Fault Detection, Isolation and Recovery

SpaceFibre provides automatic fault detection, isolation and recovery. When a fault occurs on a SpaceFibre link, it is detected and the erroneous or missing information resent. SpaceFibre recovers from intermittent faults very rapidly, detecting faults, recovering and resending data faster than SpaceWire disconnects and reconnects a link. The retry mechanism does not depend on time-outs, naturally adapting to different cable delays.

Fault detection is provided by checking each 8B/10B symbol for disparity errors and invalid 8B/10B codes. SpaceFibre has selected the 8B/10B K-codes it uses to have enhanced Hamming distance from data-codes. This means that a single bit error occurring in a data-code cannot result in a valid K-code used by SpaceFibre. In addition each data frame, broadcast frame, FCT, ACK and NACK are protected by a CRC.

Fault isolation is provided at various levels in SpaceFibre. AC coupling is used in the physical layer to prevent damage from faults that cause DC voltages exceeding the maximum permitted to appear on the transmitter outputs or receiver inputs. This feature also enables galvanic isolation to be implemented readily. At the Quality level SpaceFibre provides time containment, containing errors in the data frame in which they occur, and bandwidth containment, containing errors to the virtual channel in which they occur; an error in one VC does not affect data flowing in another VC. Babbling idiots are contained using the QoS mechanism described above.

Fault recovery is provided at the link level using a retry mechanism that resends data frames, broadcast frames and FCTs. The retry is very fast, uses a minimum amount of buffer memory, and adapts automatically to different link lengths. In addition to the retry mechanism the multi-lane functionality includes graceful degradation on lane failure. If a lane fails permanently, so that a retry or re-initialisation does not recover lane operation, a multi-lane system will continue using the remaining lanes available. This reduces the bandwidth available but does not stop the link operating. For critical operations an extra lane can be included and the graceful degradation facility will then provide automatic replacement of a faulty lane. The bit error rate (BER) of a lane is monitored and a lane reported as faulty if the (BER) is above a level which results in the effective link bandwidth being unusable. This feature allows lanes that can re-initialise successfully but which will not run for very long before having to re-initialise again, to be detected, isolated and replaced by a fully functional lane.

V. SpaceFibre Implementation, Test and Validation

The SpaceFibre standard has been written by the University of Dundee for ESA and has been widely reviewed by the international spacecraft engineering community. It has also been simulated and implemented in several forms. SpaceFibre is currently being integrated into several third party beta test applications to help refine the standard. The following sub-sections describe the various activities relating to the definition, implementation and validation of SpaceFibre.

A. Standard specification

The University of Dundee designed the lane layer of SpaceFibre with funding from ESA under the SpaceFibre contract, and the QoS and FDIR layer with funding from the European Commission (EC) SpaceWire-RT grant. The physical, multi-lane and management layers are currently being specified with ESA funding under the SpaceWire Demonstrator contract. At the moment of writing the current version of the SpaceFibre standard specification is draft F3, which is available from the SpaceWire Working Group website [1]. Work on the formal European Cooperation for Space Standardization (ECSS) standard for SpaceFibre is scheduled to start in the summer of 2014, once the technical specification is complete.

B. Simulation

As SpaceFibre was being designed by the University of Dundee, various alternative designs were simulated and traded-off with ad hoc software implementations. The aim of this simulation work was to rapidly explore alternative designs rather than provide formal validation of the standard specification.

During the SpaceWire-RT EC Framework 7 project, St. Petersburg University of Aerospace Instrumentation, performed extensive simulation of the SpaceFibre standard using SML to validate the available levels of the SpaceFibre specification and System-C to explore network level attributes of SpaceFibre. Feedback on issues and errors in the SpaceFibre specification was given for drafts C, D and E [5].

As part of the current ESA SpaceFibre Demonstrator activity Thales Alenia Space France (TAS-F) will be developing an OpNet simulation of the entire SpaceFibre protocol stack and validating its operation and the standard specification.

C. SpaceFibre IP core

In parallel with specifying the SpaceFibre standard the University of Dundee designed and tested a VHDL IP core for SpaceFibre. This was necessary to ensure that an implementation of SpaceFibre was efficient in terms of both performance and gate count. The SpaceFibre IP core was implemented in an FPGA to help test and validate the SpaceFibre specification. The IP core was used at all stages of the draft specification to validate and prove the concepts being explored. As a consequence, the VHDL IP core has gone through as many iterations as the SpaceFibre specification. At present the VHDL IP core implements all layers of the SpaceFibre specification with the exception of the Multi-Lane layer.

Within the EC SpaceWire-RT project one of the partners, ELVEES a Russian leading chip design company, explored the feasibility of implementing the SpaceFibre in a radiation tolerant ASIC. Implementation feasibility was confirmed and the size of the IP core on various chip technologies was estimated.

D. STAR Fire

To support the testing of SpaceFibre a suitable test platform was required, so STAR-Dundee Ltd. developed the STAR Fire unit. A block diagram of this unit is illustrated in Fig. 5.

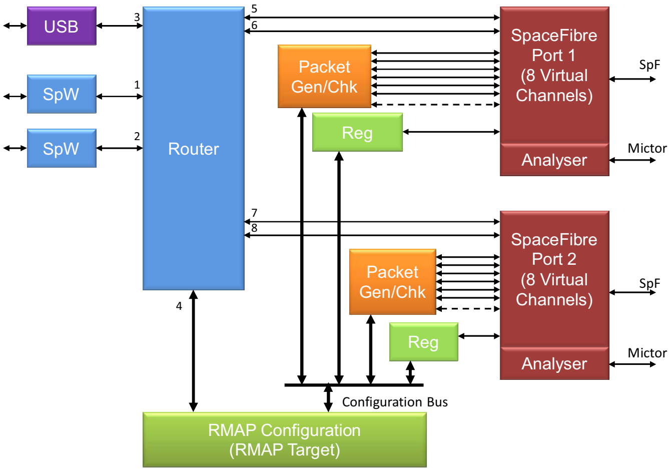
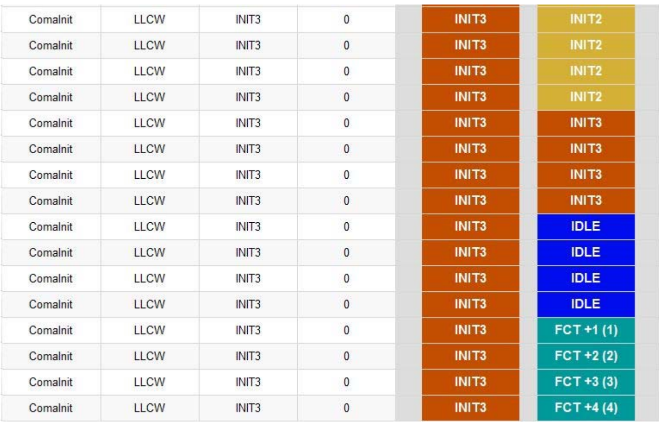

USB
SpW
SpW
Router
3
1
2
4
5
6
7
8
Packet Gen/Chk
Reg
SpaceFibre Port 1 (8 Virtual Channels)
Analyser
SpF
Mictor
Packet Gen/Chk
Reg
SpaceFibre Port 2 (8 Virtual Channels)
Analyser
SpF
Mictor
Configuration Bus
RMAP Configuration (RMAP Target)

Fig. 5. STAR Fire SpaceFibre Development Kit

The STAR-Fire unit contains two SpaceFibre interface each with eight virtual channels. Two virtual channels of each SpaceFibre interface are connected to a SpaceWire router, which also has two SpaceWire ports, a USB port and an RMAP configuration port. This allows the two SpaceWire interfaces and the USB interface to send packets through either SpaceFibre interface. To test the SpaceFibre interface at full speed and to exercise and validate the bandwidth reservation, priority and scheduled qualities of service, a packet generator and checker is attached to six of the virtual channels of each SpaceFibre interface. The STAR Fire unit is configured and controlled by a Remote Memory Access Protocol (RMAP) interface attached to the SpaceWire router. This allow configuration to be performed over the SpaceWire interfaces, USB interface or the SpaceFibre interfaces. Each SpaceFibre interface has an analyser attached which can be used to record and analyse the operation of the SpaceFibre interface. This was a very important capability during testing of the SpaceFibre IP core and validation of the standard. A graphical user interface provides access to all the capabilities of STAR Fire. Part of an example analysis display is shown in Fig. 6. where the control words being exchanged in each direction are shown in colour and the four symbols that make up the left hand control code being shown in black and white. This particular display is showing the final stages of lane initialisation resulting in FCTs being sent by the right hand side which permit the left hand side to start sending data.



ComaInit	LLCW	INIT3	0	INIT3	INIT2
ComaInit	LLCW	INIT3	0	INIT3	INIT2
ComaInit	LLCW	INIT3	0	INIT3	INIT2
ComaInit	LLCW	INIT3	0	INIT3	INIT2
ComaInit	LLCW	INIT3	0	INIT3	INIT3
ComaInit	LLCW	INIT3	0	INIT3	INIT3
ComaInit	LLCW	INIT3	0	INIT3	INIT3
ComaInit	LLCW	INIT3	0	INIT3	INIT3
ComaInit	LLCW	INIT3	0	INIT3	IDLE
ComaInit	LLCW	INIT3	0	INIT3	IDLE
ComaInit	LLCW	INIT3	0	INIT3	IDLE
ComaInit	LLCW	INIT3	0	INIT3	IDLE
ComaInit	LLCW	INIT3	0	INIT3	FCT +1 (1)
ComaInit	LLCW	INIT3	0	INIT3	FCT +2 (2)
ComaInit	LLCW	INIT3	0	INIT3	FCT +3 (3)
ComaInit	LLCW	INIT3	0	INIT3	FCT +4 (4)

Fig. 6. STAR Fire Analysis Display

E. VHiSSI

A radiation tolerant SpaceFibre interface device is being developed by University of Dundee and several partners within the Very High Speed Serial Interface (VHiSSI) European Commission Framework 7 project [6]. The VHiSSI project is integrating a complete SpaceFibre protocol engine, together with the physical layer interfaces, in a radiation tolerant chip manufactured by a European foundry. A block diagram of this device is shown in Fig. 7.

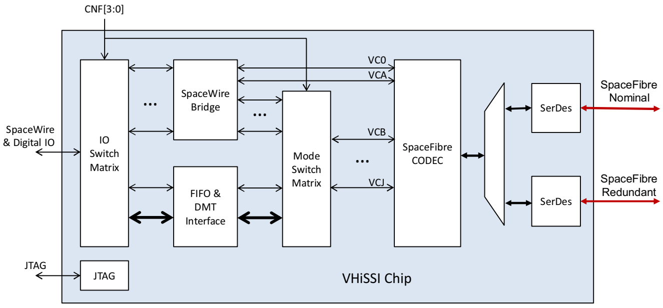


Fig. 7. VHiSSI Chip Block Diagram

There are five main functions within the VHiSSI chip:

- SpaceWire Bridge
- FIFO, DMA, Memory and Transaction Interface
- SpaceFibre Interface
- SerDes
- IO Switch Matrix
- Mode Switch Matrix

The SpaceWire Bridge provides a bridge between SpaceWire and SpaceFibre with up to 11 SpaceWire interfaces being available. The SpaceWire Bridge includes a SpaceWire router which allows routing between the SpaceWire ports and the Virtual Channel (VC) buffers of the two SpaceFibre interfaces. Configuration of the VHiSSI chip can be carried out over any SpaceWire interface connected to the embedded SpaceWire router or over VC0 or VCA of the SpaceFibre interface. The SpaceWire Bridge is connected to the IO Switch Matrix and to the Mode Switch Matrix.

The FIFO and DMA, Memory and Transaction (DMT) Interface provides various types of parallel interface into the VHiSSI chip for sending and receiving data over the SpaceFibre interfaces. The various parallel interface functions have been designed with specific application scenarios in mind and between them are able to operate with many types of local host system, including FPGAs and processors. The parallel interface is also designed to use a small number of pins, so that the VHiSSI chip can fit into a small (100 pin) package

The SpaceFibre Interface has 11 virtual channels. VC 0 is intended primarily for VHiSSI device and local system configuration and monitoring and is connected to the embedded SpaceWire router. The other VCs have programmable VC numbers and so are referred to by letters. VCA is connected to the embedded SpaceWire router. The other VCs are either connected to the SpaceWire router, directly to a SpaceWire interface, or to the parallel interface, depending on the mode of operation. Each VC supports full SpaceFibre QoS which can be configured independently for each VC. VC0 and VCA are directly connected to the embedded SpaceWire router. The other SpaceFibre VC buffers are connected to the Mode Switch Matrix which connects them to either the SpaceWire Bridge or the parallel interface. The other side of the SpaceFibre interface is connected via a multiplexer to either the nominal or redundant SerDes and CML transceiver.

The SerDes converts parallel data words from the SpaceFibre interface into a serial bit stream and vice versa. On the receive side the bit clock is recovered from the serial bit stream by the SerDes. The SerDes includes integral CML transceivers.

The IO Switch Matrix connects either the SpaceWire LVDS, SpaceWire LVTTL or parallel interface signals from the FIFO and DMT interface to the digital IO pins of the VHiSSI chip. Configuration is static and determined on exit from device reset.

The Mode Switch Matrix connects either the SpaceWire Bridge or FIFO and DMT interface (parallel interface) to the VC buffers of the two SpaceFibre interfaces. Configuration is static and determined on exit from device reset.

F. Electrical Connectors and Cable Assemblies

Axon’s AxoMach range of connectors and cables are currently baselined for use within SpaceFibre although the specification will allow the use of other types of connector and cable. Within the frame of the ESA SpaceFibre Demonstrator activity Axon will investigate alternative connector configurations and provide an open specification for the cable and connectors. The spaceflight capable AxoMach cable assembly is illustrated in Fig. 8. Laboratory testing has shown that SpaceFibre will operate at a distance of over 5 m using these cables and connectors.

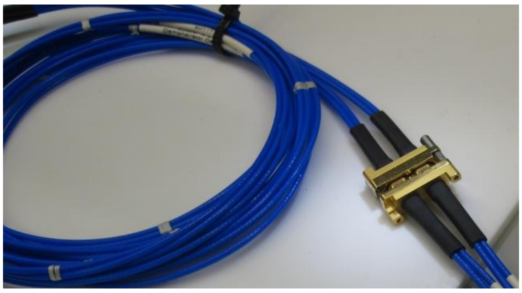


Fig. 8. Prototype SpaceFibre Flight Cable from Axon

G. Fibre Optic Transceivers and Cable Assemblies

For longer distances Patria are working for ESA on fibre-optic cables, connectors and transceivers. Tests have been made on prototype fibre-optic devices. In Fig. 9. two STAR Fire units are being used to test SpaceFibre signals running through radiation tolerant fibre optic transceivers and over 100 m fibre optic cable.

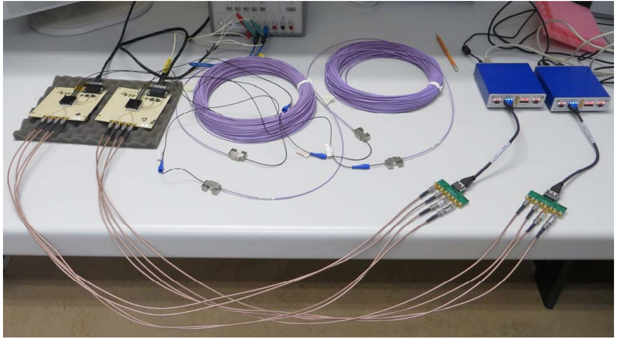


Fig. 9. STAR Fire Testing 100m Fibre Optic Cable

H. Beta Test Sites

Several ESA projects are using the Dundee SpaceFibre IP core under a Beta evaluation programme including:

- High Processing Power DSP, Astrium and STAR-Dundee Ltd.
- High Performance COTS Based Computer, Astrium and CGS.
- Leon with Fast Fourier Transform Co-processor, SSBV.
- FPGA Based Generic Module and Dynamic Reconfigurator, Bielefeld University.
- Next Generation Mass Memory, Astrium, IDA and University of Dundee.

Feedback from these beta sites will be used to improve the SpaceFibre standard and the SpaceFibre VHDL IP core and related documentation.

In additional some other organisations are currently testing the Dundee SpaceFibre VHDL IP core for space-based instrumentation interfacing applications.

I. Japanese Implementations

NEC and Melco are both developing SpaceFibre interface devices to the specification produced by the University of Dundee. This work has provided valuable feedback on the specification and implementation of SpaceFibre. Interoperability testing in December 2012 and April 2013 has been successful with various levels of the SpaceFibre protocol stack being implemented and tested. Several issues were uncovered during the interoperability testing which resulted in clarifications and errors being resolved in the SpaceFibre standard specification.

J. Spaceflight Engineering Model

To raise the TRL of SpaceFibre a spaceflight engineering model is being developed by Astrium in the frame of the ESA SpaceFibre Demonstrator project. This will use the Dundee IP core integrated with some other interface and test logic in a Microsemi AX2000 FPGA. Texas Instrument TLK2711 devices will be used for the serialisation and de-serialisation. Both devices are available in radiation tolerant, space grade parts. A block diagram of the planned SpaceFibre engineering model is illustrated in Fig. 10. Extensive testing, including EMC testing, will be carried out with this board in 2014.

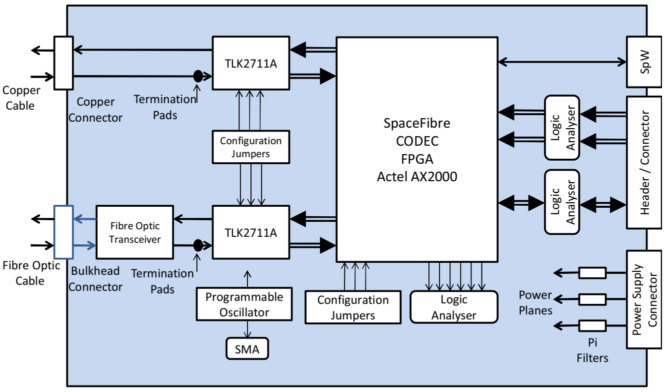
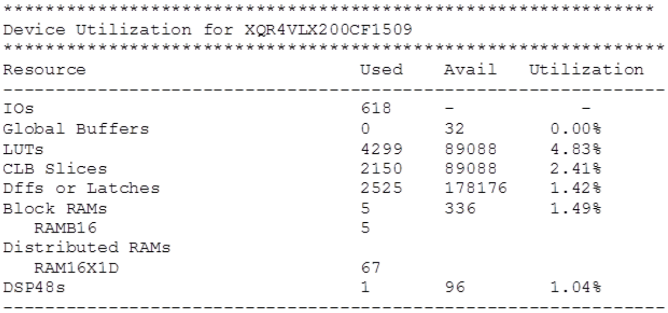


Fig. 10. SpaceFibre Engineering Model

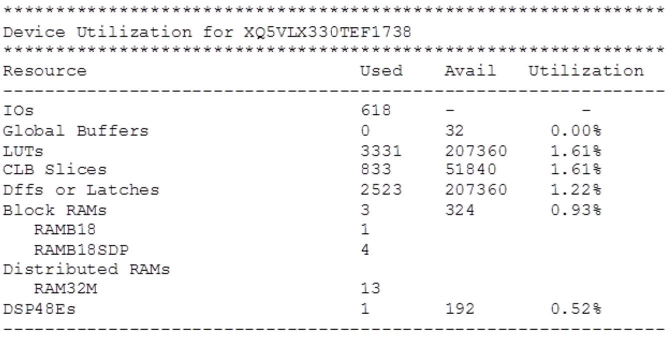
VI. FPGA Resources

The FPGA resources required for a SpaceFibre link with a single virtual channel are detailed for various types of space qualified, radiation tolerant FPGAs in Fig. 11. to Fig. 13. The utilisation for an 8 virtual channel interface is about twice that of a single virtual channel interface.



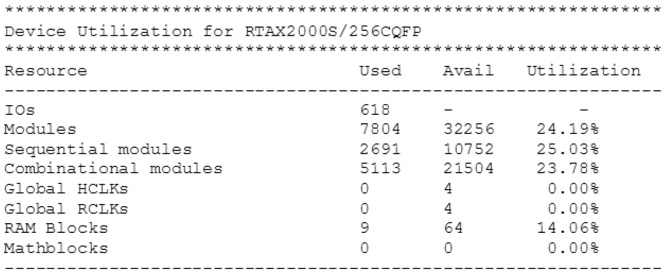
```
***************************************************************
Device Utilization for XQR4VLX200CF1509
****************************************************************
Resource                          Used    Avail   Utilization
----------------------------------------------------------------
IOs                               618     -            -
Global Buffers                    0       32         0.00%
LUTs                              4299    89088      4.83%
CLB Slices                        2150    89088      2.41%
Dffs or Latches                   2525    178176     1.42%
Block RAMs                        5       336        1.49%
   RAMB16                         5
Distributed RAMs
   RAM16X1D                       67
DSP48s                            1       96         1.04%
----------------------------------------------------------------
```

Fig. 11. SpaceFibre Single Virtual Channel Xilinx Virtex 4 FPGA Utilisation



```
****************************************************************
Device Utilization for XQ5VLX330TEF1738
****************************************************************
Resource                          Used    Avail    Utilization
----------------------------------------------------------------
IOs                               618     -             -
Global Buffers                    0       32        0.00%
LUTs                              3331    207360    1.61%
CLB Slices                        833     51840     1.61%
Dffs or Latches                   2523    207360    1.22%
Block RAMs                        3       324       0.93%
   RAMB18                         1
   RAMB18SDP                      4
Distributed RAMs
   RAM32M                         13
DSP48Es                           1       192       0.52%
----------------------------------------------------------------
```

Fig. 12. SpaceFibre Single Virtual Channel Xilinx Virtex 4 FPGA Utilisation



```
***************************************************************
Device Utilization for RTAX2000S/256CQFP
***************************************************************
Resource                          Used    Avail    Utilization
---------------------------------------------------------------
IOs                               618     -             -
Modules                           7804    32256     24.19%
Sequential modules                2691    10752     25.03%
Combinational modules             5113    21504     23.78%
Global HCLKs                      0       4          0.00%
Global RCLKs                      0       4          0.00%
RAM Blocks                        9       64        14.06%
Mathblocks                        0       0          0.00%
---------------------------------------------------------------
```

Fig. 13. SpaceFibre Single Virtual Channel Xilinx Virtex 4 FPGA Utilisation

VII. Conclusions

SpaceFibre is a multi-gigabit/s networking technology designed specifically for spaceflight applications. It incorporates a comprehensive quality of service capability providing integrated bandwidth reservation, priority and scheduling. Efficient, effective and rapid fault detection, isolation and recovery mechanisms are included in the SpaceFibre interface, enabling rapid detection and recovery from link level errors.

SpaceFibre supports high data-rate payloads, for example 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 onboard network technology to be used across many different mission applications resulting in cost reduction and design reusability. SpaceFibre uses a packet format which is the same as SpaceWire enabling simple connection between existing SpaceWire equipment and high-speed SpaceFibre links and networks. It reduces development time and costs, because of its integrated QoS and FDIR capabilities and because it simplifies previously complex onboard data-handling architectures.

Acknowledgment

The research leading to these results has received funding the European Space Agency under ESA contract numbers 4000102641 and 17938/03/NL/LvH from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement numbers 263148 and 284389. We would also like to thank Martin Suess the ESA project manager for the SpaceFibre related activities for his help, advice and guidance.

References

[1] S. Parkes, A. Ferrer and A. Gonzalez, “SpaceFibre Standard”, Draft F3 September 2013, available from http://space-env.esa.int/indico/confLogin.py?confId=32 (last accessed 15th Feb 2014).

[2] S. Parkes, A. Ferrer, A. Gonzalez, & C. McClements, “SpaceFibre: Multiple Gbits/s Network Technology with QoS, FDIR and SpaceWire Packet Transfer Capabilities”, International SpaceWire Conference, Gothenburg, June 2013.

[3] S. Parkes, “Never Mind the Quality, Feel the Bandwidth: Quality of Service Drivers for Future Onboard Communication Networks”, paper no. IAC-10.B2.6.6, 61st International Astronautical Congress, Prague 2010.

[4] ECSS Standard ECSS-E-ST-50-12C, “SpaceWire, Links, Nodes, Routers and Networks”, Issue 1, European Cooperation for Space Data Standardization, July 2008.

[5] V. Olenev, I. Lavrovskaya, and I. Korobkov, “SpaceWire-RT/SpaceFibre Specification and Modelling”, International SpaceWire Conference, Gothenburg, 2013.

[6] S. Parkes, A. Ferrer, A. Gonzalez, C. McClements, R. Ginosar, T Liran, G Sokolov, G Burdo, N Blatt, P Rastetter, M Krstic, A Crescenzio, “A Radiation Tolerant SpaceFibre Interface Device”, International SpaceWire Conference, Gothenburg, 2013.