SpaceFibre: A Multi-Gigabit/s Interconnect for Spacecraft Onboard Data Handling

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

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

Abstract— **SpaceFibre is a spacecraft onboard data link and network technology being developed by University of Dundee for the European Space Agency (ESA), which runs over both copper and fibre optic cables. Initially targeted at very high data rate payloads such as Synthetic Aperture Radar (SAR) and multi-spectral imaging instruments, SpaceFibre is capable of fulfilling a wider set of spacecraft onboard communications applications because of its inbuilt QoS and FDIR capabilities and its backwards compatibility with the ubiquitous SpaceWire technology. SpaceFibre operates at 2.5 Gbits/s providing 12 times the throughput of a SpaceWire link with current flight qualified technology and allowing data from multiple SpaceWire devices to be concentrated over a single SpaceFibre link. This substantially reduces cable harness mass and simplifies redundancy strategies. The innovative QoS mechanism in SpaceFibre provides concurrent bandwidth reservation, priority and scheduled QoS. This simplifies spacecraft system engineering through integrated quality of service (QoS), which reduces system engineering costs and streamlines integration and test. Novel integrated FDIR support provides galvanic isolation, transparent recovery from transient errors, error containment in virtual channels and frames, and "Babbling Idiot" protection. SpaceFibre enhances onboard network robustness through its inherent FDIR and graceful degradation techniques incorporated in the network hardware. This simplifies system FDIR software, reducing development and system validation time and cost. SpaceFibre includes low latency event signalling and time distribution with broadcast messages. This enables a single network to be used for several functions including: transporting very high data rate payload data, carrying SpaceWire traffic, deterministic delivery of command/control information, time distribution and event signalling. SpaceFibre is backwards compatible with existing SpaceWire equipment at the packet level allowing simple interconnection of SpaceWire devices into a SpaceFibre network and enabling that equipment to take advantage of the QoS and FDIR capabilities of SpaceFibre.**

Index Terms—**SpaceWire, SpaceFibre, Network, Spacecraft Onboard Data-Handling, Quality of Service, FDIR, Next Generation Interconnect.**

TABLE OF CONTENTS

1. INTRODUCTION

SpaceFibre [1][2][3] is a multi-Gigabit/s data link and network technology specifically designed for spaceflight applications, which is being developed by ESA to support high data rate payload data-handling such as synthetic aperture radar (SAR), multi-spectral imaging systems and fast mass memory units. The draft SpaceFibre standard has been written by the University of Dundee for ESA and has been 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 finalise the standard in preparation for formal standardisation by the European Cooperation for Space Standardization (ECSS) in 2015. SpaceFibre is an open standard currently available from the ESA SpaceWire Working Group website.

SpaceFibre runs over both electrical and fibre-optic media. It includes built in, very efficient, quality of service (QoS) and fault detection, isolation and recovery (FDIR) techniques, which simplify the use of SpaceFibre enormously; providing substantial system level benefits without requiring the implementation of complex performance limiting software protocols.

SpaceFibre is backwards compatible with SpaceWire [4] at the packet level allowing easy bridging between SpaceWire and SpaceFibre, so that existing SpaceWire devices can be incorporated into a SpaceFibre network and take advantages of its performance and QoS and FDIR capabilities.

While SpaceFibre has been targeted initially at spacecraft onboard payload data-handling applications it has also been designed to provide a fully integrated spacecraft network which:

- Supports high data rate payloads,
- Carries data from existing SpaceWire instruments,
- Performs low latency time-distribution,
- Provides low latency event signalling,
- Provides deterministic data delivery to support guidance and navigation control applications

SpaceFibre has been designed, reviewed and validated through analysis, simulation and hardware implementation, in a series of stages with feedback from each validation cycle feeding into the design. This has resulted in a mature well tested standard which will be released to ECSS for formal standardization at the end of 2014. The Technology Readiness Level (TRL) is already at TRL 5 with an implementation designed in flight proven radiation tolerant FPGA and SerDes devices. It will be raised to TRL 6 with application demonstrations in the near future. An experimental radiation tolerant SpaceFibre interface has been designed and manufactured and is currently undergoing tests. It is an open standard designed specifically for the next generation of spacecraft onboard network.

The SpaceFibre standard is described in section 2. of this paper. The QoS mechanism is described in section 3. and the FDIR in section 4. . The design of a SpaceFibre IP core is outlined in section 5. and important test equipment used for testing SpaceFibre described in section 6. . Results of tests on the QoS and FDIR functions are described in section 7. . An experimental ASIC implementation of SpaceFibre is described in section 8. . The ways in which the SpaceFibre standard has been validated are explained in section 9. . Finally conclusions are made in section 10. .

2. THE SPACEFIBRE STANDARD

SpaceFibre is currently a mature draft standard being specified by the University of Dundee with contributions from several other organisations. The protocol stack for SpaceFibre is illustrated in Figure 1.

The network layer protocol provides two services for transferring application information over a SpaceFibre network; the packet transfer service and the broadcast message service. The Packet Transfer Service transfers SpaceFibre packets over the SpaceFibre network, using the same packet format and routing concepts as SpaceWire. The broadcast message service broadcasts short messages carrying time and synchronisation information to all nodes on the network.

The QoS and FDIR layer provides quality of service and flow control for 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. It also provides a retry capability; detecting any frames or control codes that go missing or arrive containing errors and resending them.

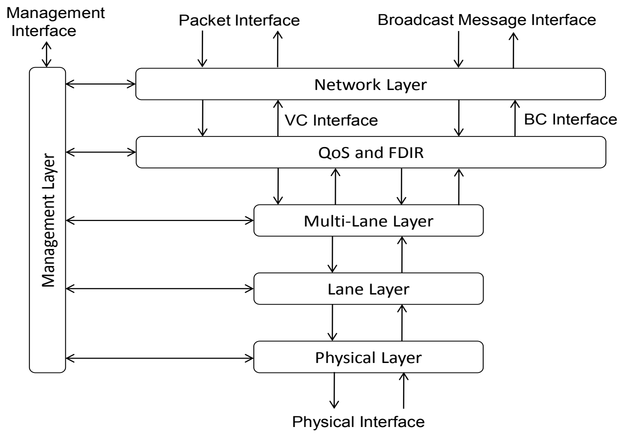

Figure 1 - SpaceFibre Protocol Stack

The Multi-Lane layer is able to operate 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. It does this rapidly without any external intervention.

The Lane layer initialises each lane and re-initialises any lane that detects an error. Data is encoded into symbols for transmission using 8B/10B encoding and decoded in the receiver. 8B/10B codes are DC balanced supporting AC coupling of SpaceFibre interfaces.

The Physical layer serialises the 8B/10B symbols and sends 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.

The management layer supports the configuration, control and monitoring of all the layers in the SpaceFibre protocol stack.

The SpaceFibre standard has been simulated, implemented and reviewed at all stages of its research, design and development. The lane and QoS layers of SpaceFibre are fully defined and have been extensively tested with simulations by at least three independent organisations, and by implementation in FPGAs. The physical layer is well on the way to being complete with final inputs on eye pattern masks etc. to be added. The multi-lane layer has been designed and simulated, and is currently in the process of being implemented and tested in FPGAs. Once this testing is complete and the specification updated to resolve any issues found, draft G of the SpaceFibre standard will be issued and an ECSS working group will be convened to finalise the standard for formal approval.

The SpaceFibre network layer will be a separate standard document. The network layer uses the same packet format as SpaceWire and supports path and logical addressing.

3. 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. An N-Char is a data byte, End of Packet marker (EOP) or Error End of Packet marker (EEP). 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.

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

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 from and to the application. 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.

Priority is one example of a QoS attribute. Other types of QoS are considered in the subsequent sections.

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. Precedence is derived from the bandwidth reserved for a virtual channel and its priority, as described in the following sections.

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 impose 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 gradually 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 Percentage. When a frame of data is sent by any virtual channel, each virtual channel computes the amount of bandwidth that it would have been permitted to send in the time interval that the last frame was sent. This is known as the Bandwidth Allowance. Bandwidth Allowance is calculated as follows:

$$\textit{Bandwidth Allowance} = \textit{Expected BW\% x Last Frame Bandwidth} \quad (1)$$

Where Expected BW% is the portion of overall link bandwidth that a virtual channel wishes to use, and Last Frame Bandwidth is the amount of data sent by any virtual channel in the last data frame. Each virtual channel can use its Bandwidth Allowance to determine its Bandwidth Credit, which is effectively the amount of data it can send and still remain within its Expected Bandwidth. Bandwidth Credit for a particular virtual channel is the amount of data that the virtual channel is permitted to send minus the amount of data it has actually sent, i.e. the Bandwidth Allowance less the Bandwidth Used accumulated over time.

Bandwidth Credit is calculated for each virtual channel as follows:

$$BandwidthCredit = \sum_{Frames} \frac{BandwidthAllowance - UsedBandwidth}{ExpectedBandwidth} \quad (2)$$

Where Used Bandwidth is the amount of data sent by a particular virtual channel in the last data frame, which is zero for all virtual channels except for the one that sent the last frame.

Consider the following example. A virtual channel is allocated 10% of the link bandwidth (Expected Bandwidth = 0.1). Each frame being sent contains 250 bytes, so the Bandwidth Allowance is 0.1x250 =25 bytes/unit of time. If we consider the summation over say 20 frames and in that interval 2 frames are sent by the virtual channel of interest, the summed Bandwidth Allowance is 20*25 bytes/unit of time, the summed Used Bandwidth is 2x250 bytes/unit of time. So the Bandwidth Credit is then (20*25 – 2*250)/0.1 = 0 bytes/unit of time which is as expected since the virtual channel sent 2 out of 20 frames which is 10%. If the virtual channel sends 1 frame out of 20 then the Bandwidth Credit is (20*25-1*250)/0.1 = +2500 bytes/unit of time so the Bandwidth Credit increases significantly and it is more likely that the virtual channel will be permitted to send the next frame. If the virtual channel sends 3 frames out of 20 then the Bandwidth Credit is (20*25-3*250)/0.1 = -2500 bytes/unit of time so the Bandwidth Credit drops significantly and it is less likely that the virtual channel will be permitted to send the next frame.

The Bandwidth Credit is updated every time a data frame for any virtual channel has been sent. A Bandwidth Credit value close to zero indicates nominal use of bandwidth by the virtual channel. A negative value indicates that the virtual channel is using more than its expected amount of link bandwidth. A positive value indicates that the virtual channel is using less than its expected amount of link bandwidth.

To simplify the hardware required to calculate the Bandwidth Credit it is allowed to saturate at plus or minus a Bandwidth Credit Limit, i.e. if the Bandwidth Credit reaches a Bandwidth Credit Limit it is set to the value of the Bandwidth Credit Limit.

When the Bandwidth Credit for a virtual channel reaches the negative Bandwidth Credit Limit it indicates that the virtual channel is using more bandwidth than expected. This may be recorded in a status register and used to indicate a possible error condition. A network management application is able to use this information to check correct utilisation of link bandwidth by its various virtual channels.

For a virtual channel supporting bandwidth reserved QoS, the value of the bandwidth counter provides the precedence value for that virtual channel.

The operation of a bandwidth credit counter is illustrated in Figure 2.

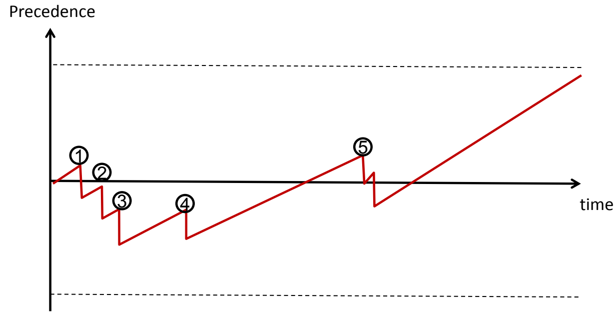

Figure 2 - Bandwidth Credit Counter

The bandwidth credit for a particular VC increments gradually. At point (1) a frame is sent from by this VC, resulting in a sudden drop in credit. The size of the drop is amount of data sent in the frame divided by the percentage bandwidth reserved for the VC. This means that the smaller the percentage bandwidth the larger the drop, and hence the longer it takes to regain bandwidth credit.

After the drop at point (1) the bandwidth credit gradually increments until point (2) when another frame is sent by the VC. Further frames are sent at points (3), (4), (5) etc. If the frames sent are full frames then the drop in bandwidth credit every time a frame is sent, will be the same size.

The bandwidth credit counter for another VC is illustrated in Figure 3. This VC has about half the bandwidth of the VC in Figure 2 allocated to it. This means that the drops in bandwidth credit when frames are sent by this VC are about twice the size, as can be seen Figure 3 at points (1), (2) and (3).

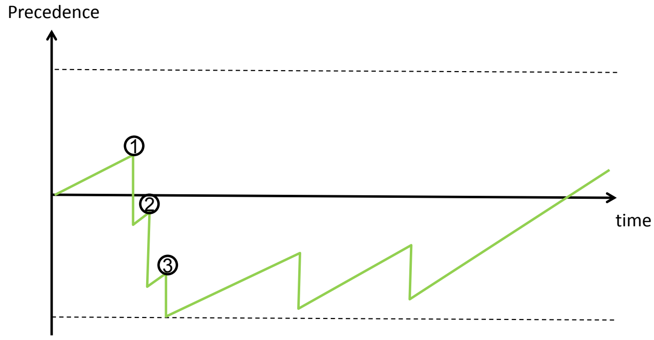

Figure 3 - Bandwidth Credit Counter with Smaller Reserved Bandwidth

The bandwidth credit counter of another VC is shown in Figure 4. In this case the bandwidth credit slowly increments and although some frames are sent at points (1), (2) and (3), the bandwidth credit eventually saturates, reaching its maximum permitted value at point (4). Although more bandwidth should be accumulated after point (4) this is effectively ignored since the maximum possible bandwidth credit has been reached. At point (5) a frame is sent once more, resulting in a drop from the maximum bandwidth credit value.

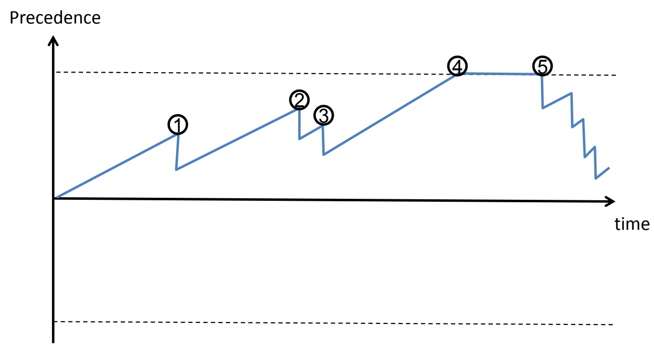

Figure 4 - Bandwidth Credit Counter Reaching Saturation

All three VCs are shown together in Figure 5. 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 bandwidth credit 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 bandwidth credit count. After this the red VC is allowed to send a further data frame at point (8).

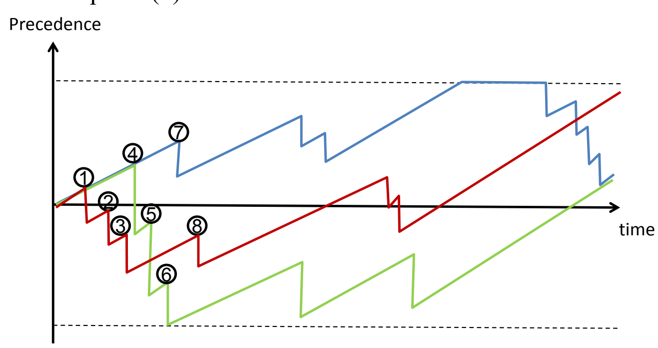

Figure 5 - Bandwidth Credit of Competing VCs

If the bandwidth credit counter reaches the minimum possible bandwidth credit 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.

Similarly if the bandwidth credit counter stays at the maximum possible bandwidth credit 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 Bandwidth Credit for different values of Expected Bandwidth is illustrated in Figure 6.

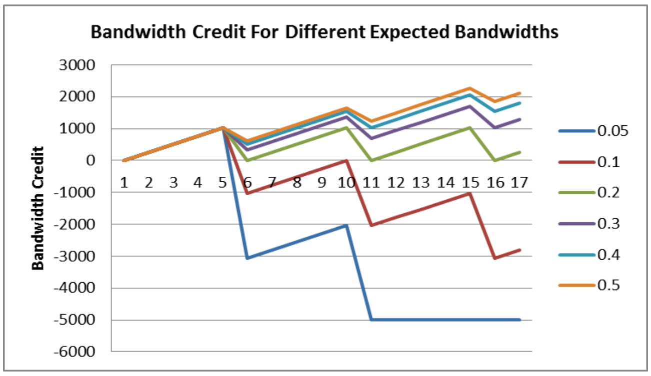

Figure 6 - Bandwidth Credit for Different Expected Bandwidths

Along the x-axis is time expressed as transmitted frame number. It is assumed in the diagram that all frames are the same size. In frame numbers 5, 10 and 15 a frame is sent by the particular virtual channel being considered. Each line in the chart shows what happens to the Bandwidth Credit for different values of the Expected Bandwidth. Consider the green line in the middle (Expected BW = 0.2). The Bandwidth Credit slowly increases until the virtual channel sends a frame at time 5. The Bandwidth Credit then drops back to zero. If the Expected Bandwidth was higher the drop in Bandwidth Credit would be less and if the Expected Bandwidth was lower the drop in Bandwidth Credit would be more. The blue line at the bottom (Expected BW = 0.05) drops by 4000 units each time a frame is sent eventually saturating at the value of -5000. It can be seen from this diagram that if the virtual channel has a higher Expected Bandwidth its Bandwidth Credit is higher, assuming that the data the virtual channel sends is the same for each case.

The bandwidth credit value is the precedence used by the medium access controller to determine which VC is permitted to send the next data frame.

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. Figure 7 shows three priority levels. SpaceFibre has 16 priority levels.

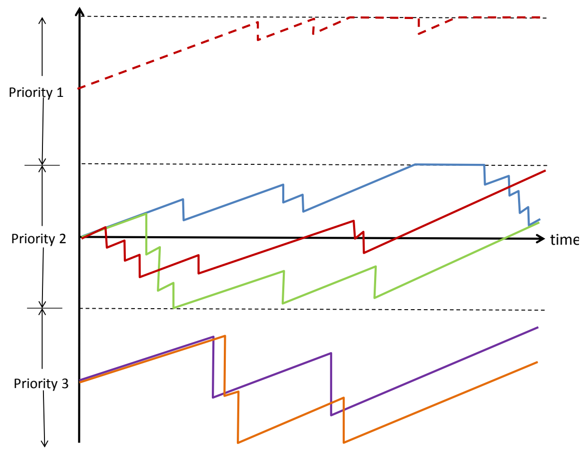

Figure 7 - Multi-Layered Precedence Priority QoS

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.

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 Figure 8.

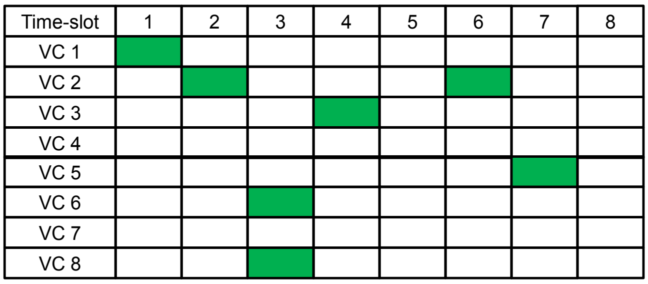

Figure 8 - Scheduled Quality of Service

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 3. The time-slot duration is a system level parameter, typically 1 ms, and there are 64 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 user does not want to use scheduling the schedule table is simply filled completely, allowing any VC to send in any time-slot, competing with other VCs using 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 Figure 8 time-slot 3 and VCs 6 and 8.

Figure 9 illustrates a very efficient way of implementing deterministic data delivery over SpaceFibre while using the maximum amount of link bandwidth for non-deterministic traffic. VCs 1 and 2 represent the deterministic traffic, for example Attitude and Orbit Control System (AOCS) and housekeeping. The AOCS has to read data from AOCS sensors and write commands to AOCS actuators every 4 ms; it does this using SpaceWire RMAP commands, for example. The deterministic VCs are allocated specific time-slots and given the highest priority. This means that when an allocated time-slot comes along the deterministic VC can send all of its data first. When it no longer has any data to send, other VCs can send data competing with each other for access to the link using precedence. The deterministic VC is not allowed to send data in other time-slots, only in the ones it has been allocated, ensuring strict time limits on the transfer of data by the deterministic VC. At the start of the allocated time-slot the deterministic VC will send its RMAP commands in a burst and will also send any RMAP replies. This is illustrated at the bottom of Figure 9, where a burst of traffic from the deterministic VC is sent at the start of a time-slot and other traffic fills up the time-slot when there is nothing else for the deterministic VC to send.

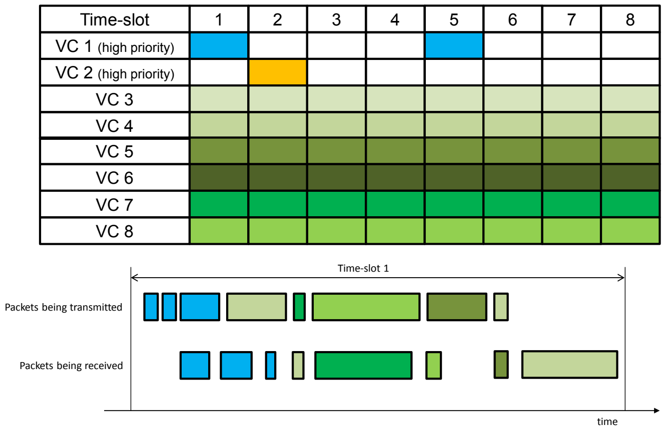

Figure 9 – Determinism with Scheduling and Priorty

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. 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, not as separate options. 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.

4. 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, Flow Control Token (FCT), Acknowledgement (ACK) and Negative Acknowledgement (NACK) are protected by a CRC. FCTs are used to manage the flow of data over a link and ACKs and NACKs are used to support the link error recovery.

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 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.

5. A SpaceFibre IP Core

A SpaceFibre IP core has been designed and developed by University of Dundee and STAR-Dundee to test and validate the SpaceFibre specification. This has been updated and used to re-validate each revision of the SpaceFibre standard. A block diagram showing the interfaces to the IP Core is given in Figure 10. The current version SpaceFibre IP core is compliant to the draft F3 version of the SpaceFibre standard and supports all its features with the exception of multi-laning.

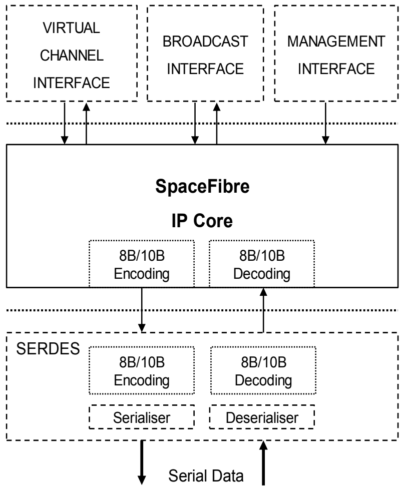

Figure 10 - SpaceFibre IP Core Interfaces

The SpaceFibre IP Core is designed to interface with an external SerDes device using the internal 8B/10B encoder/decoder or to an external 8B/10B encoder/decoder and SerDes device. This allows the SpaceFibre IP Core to be used with space qualified SerDes such as the TLK2711-SP device from Texas Instruments. The application interface to the SpaceFibre IP core comprises three separate interfaces:

1. A virtual channel interface, which is used to send and receive SpaceFibre packets over the virtual channels in the interface.
2. A broadcast interface, which is used to send broadcast messages over the SpaceFibre network.
3. A management interface, which is used to configure, control and monitor the status of the SpaceFibre interface.

The footprints of a SpaceFibre link with a single virtual channel when implemented in various types of space qualified, radiation tolerant FPGAs are listed below:

Xilinx XQR4VLX200: 5%

Xilinx XQ5VLX330: 2%

Microsemi RTAX2000S: 25%.

The utilisation for an 8 virtual channel interface is about twice that of a single virtual channel interface.

The SpaceFibre IP core has been designed to support the testing of the SpaceFibre standard. It has not been designed for speed or size. A version of the SpaceFibre IP core targeted for high performance and small size in flight-qualified FPGAs is currently being developed by STAR-Dundee Ltd. This IP core is designed to support instrument interfacing with SpaceFibre using existing flight proven FPGAs and SerDes devices.

6. SpaceFibre Test Equipment

STAR-Dundee has developed a range of SpaceFibre test and development equipment. The first unit, STAR Fire, was designed to support the testing of SpaceFibre and includes SpaceWire to SpaceFibre bridging, pattern generation and checking for multiple virtual channels and link analysis capabilities. A block diagram of STAR Fire is shown in Figure 11 and a photograph in Figure 12.

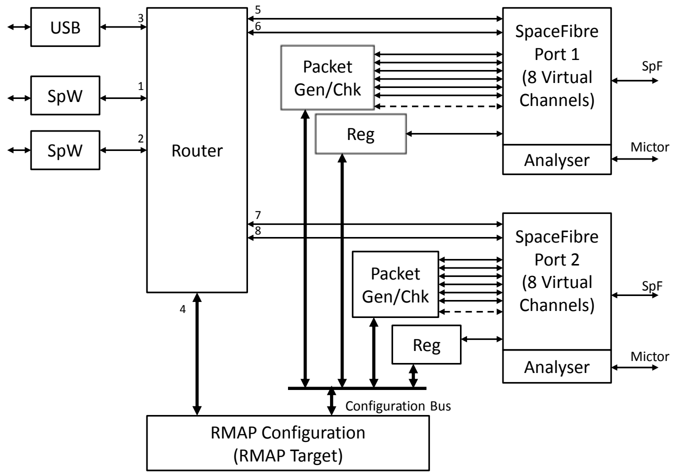

Figure 11 - STAR Fire: SpaceFibre Interface and Analyzer

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 allows configuration to be performed over the SpaceWire interfaces, USB interface or the SpaceFibre interfaces. Each SpaceFibre interface has an analyzer attached which can be used to record and analyze the operation of the SpaceFibre interface.

Figure 12 - STAR Fire Unit

A graphical user interface provides access to all the capabilities of STAR Fire. Example analysis displays are shown in Figure 13 and Figure 14. In the Word Viewer (Figure 13) the control words being exchanged in each direction are shown in colour and the four symbols that make up the control codes being shown in black and white. Time goes down the page.

Figure 13 - STAR Fire Analysis Display: Word Viewer

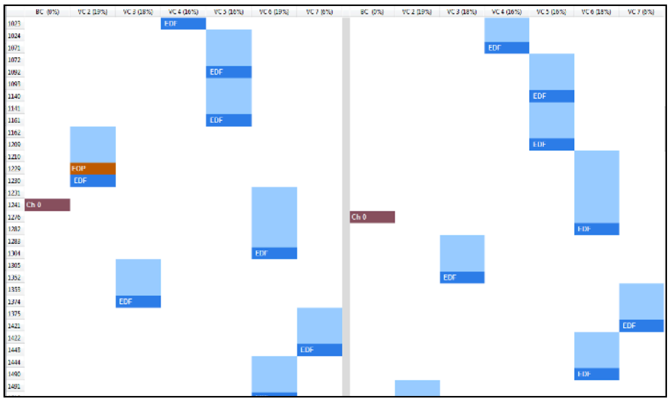

Figure 14 - STAR Fire Analysis Display: Frame Viewer

In the Frame Viewer (Figure 14) each side of the view shows the data frames flowing in each virtual channel; one half of the view shows data frames flowing in one direction and the other half shows data frames flowing in the other direction. Each column represents a virtual channel with the frames flowing through the virtual channel being shown in light blue. The end of packet markers (EOP), in darker blue, show the end of each SpaceFibre packet. A SpaceFibre frame can hold part, all, or more than one SpaceFibre packet. The broadcast channel is also shown in the leftmost column of each half of the display.

In Figure 15 the configuration and bandwidth display of the STAR Fire unit is shown. This GUI can be used to set up each of the virtual channels along with the data generator sending data over the virtual channel.

The virtual channel being configured is VC2. Its priority setting is set to the highest priority. The Bandwidth Allowance is set to 85% so that VC is permitted to use up to 85% of the link bandwidth. The time-slot setting specifies in which of 32 time-slots the VC is allowed to send data. This is done using a 32-bit pattern with a bit 0 set to 1 if the VC is allowed to send in time-slot 0 and so on. Hex is used to write the time-slot value, with ffffffff meaning that the VC can send in all time-slots. Note the STAR Fire unit currently implements 32 time-slots instead of 64. The data rate generator is set to generate packets of 1023 32-bit words at a data rate which is 82% of the full link bandwidth. The green bar shows the results of these settings with VC2 using 82% of the link bandwidth since it is of highest priority, is allocated 85% of the link bandwidth and is only capable of sending 82% of the link bandwidth.

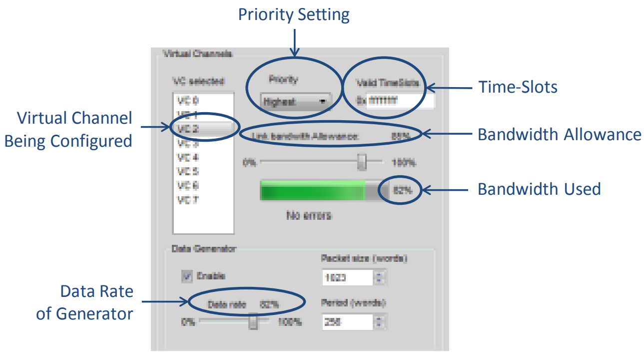

Figure 15 – STAR Fire Configuration and Bandwidth Display

A cPCI interface board has also been developed for SpaceFibre which is compatible with cPCI, RASTA and National Instruments PXI systems. This board can provide a number of different SpaceFibre functions including SpaceFibre interface, SpaceWire to SpaceFibre bridging and SpaceFibre Router functions. This board is expected to be available early in 2015. The STAR Fire unit is currently available from STAR-Dundee along with the SpaceFibre IP core.

7. TESTING THE QOS AND FDIR

The QoS and FDIR capabilities of SpaceFibre were tested using the STAR Fire unit. Figure 16 shows VC2 and VC3 configured as follows:

- VC2: highest priority, 85% bandwidth allowance and data generator bandwidth of 82%.
- VC3: low priority, 85% bandwidth allowance and data generator bandwidth of 82%.

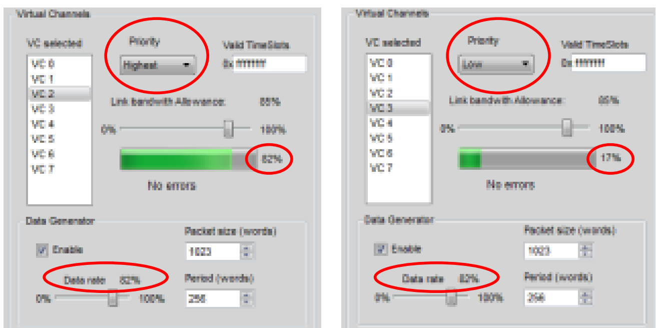

Figure 16 – STAR Fire Priority Test

Clearly, the two VCs have been allocated more than 100% of the link bandwidth, each having 85% of the link bandwidth, so one of them is not going to get its allocated share. VC2 is set to high priority and its data generator is set to generate 82% of the link bandwidth so the result shown on the green bar is that VC2 uses 82% of the link bandwidth. The remaining 17% (taking into account rounding errors, etc.) is used by the low priority VC3.

Figure 17 shows a snapshot the frames being interleaved from VC2 and VC3 with VC2 being allowed to send many more frames than VC3.

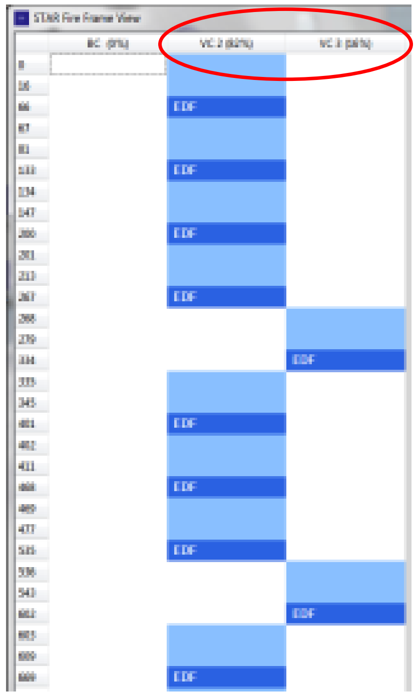

Figure 17 – STAR Fire Priority Frame Interleaving

The bandwidth reservation test configuration illustrated in Figure 18.

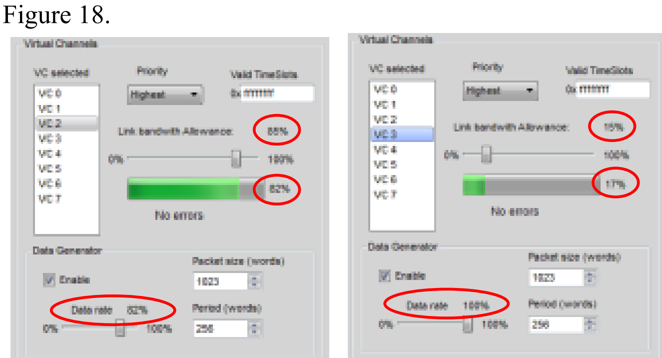

Figure 18 – STAR Fire Bandwidth Reservation Test

VC2 and VC3 configured as follows:

- VC2: highest priority, 85% bandwidth allowance and data generator bandwidth of 82%.
- VC3: highest priority, 15% bandwidth allowance and data generator bandwidth of 100%.

In this case both VCs are set to the same priority level, VC2 is given 85% bandwidth allowance and VC3 has 15% bandwidth allowance. The data generator for VC2 is generating data at 82% link bandwidth so the resulting bandwidth is 82%. This leaves 18% of the available link bandwidth for VC3. VC3 is allocated 15% bandwidth but its data generator will generator up to 100% of link bandwidth so VC3 fills the bandwidth not used by VC2, giving a bandwidth for VC3 of 18% (17% taking into account rounding errors, etc). A VC is allowed to use more than its allocated bandwidth if there is no other VC with higher precedence that wants to use that bandwidth.

The scheduling capability of SpaceFibre is illustrated in Figure 19.

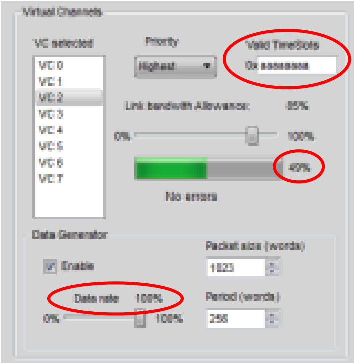

Figure 19 – STAR Fire Scheduling Test

VC 2 is the only VC being used and its data generator is set to 100%. The time-slots are set to aaaaaaaa which is a 1010… bit pattern, specifying that VC2 is able to transmit every other time-slot. The result is that VC2 sends data in bursts when its allocated time-slots arrive as shown in Figure 20. Since VC2 sends data continuously in every other time-slot its bandwidth is 50% of the overall link bandwidth (49% taking into account rounding errors, etc.).

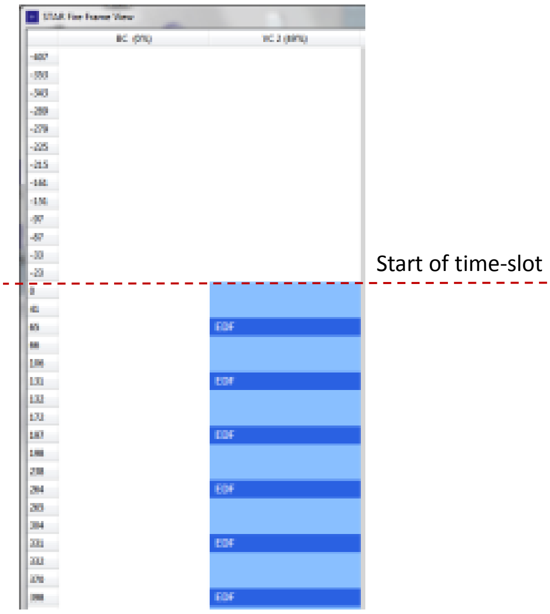

Figure 20 – STAR Fire Scheduling Frames

To test the FDIR capabilities of SpaceFibre two STAR Fire units were connected together and SpaceWire packets sent between then using a STAR-Dundee SpaceWire Brick, as illustrated in Figure 21.

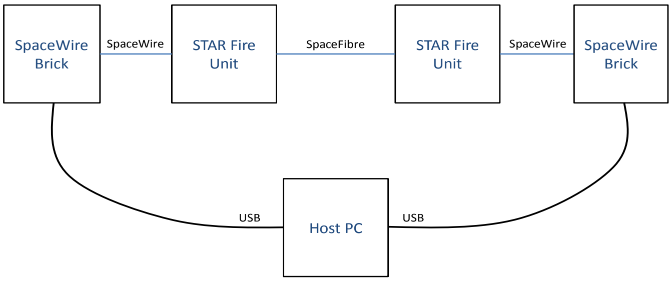

Figure 21 – STAR Fire FDIR Test Setup

A host computer generates SpaceWire packets which are sent via a SpaceWire-USB Brick which sends them to a STAR Fire unit over a SpaceWire link. The SpaceWire link is connected to VC1 inside the STAR Fire unit and the SpaceWire packets are sent over the SpaceFibre link using VC1. A STAR Fire unit at the far end of the link receives the SpaceWire packets and sends them over a SpaceWire link to a second SpaceWire-USB Brick. The received packets are passed up to the host computer and checked for correctness. The STAR Fire unit has a means for injecting bit flips into the data at a configurable bit error rate (BER). These transient errors were injected into the link at various BERs up to a BER of 10^{-6}. Each time an error was detected in a data frame or control word that information was resent. It takes of the order of 4 µs to resend the erroneous data. The SpaceWire packets cross the SpaceFibre link with no errors being reported by the packet checking software. The only effect of the high BER is a slight degradation of overall available link bandwidth to around 95%.

Error recovery in SpaceFibre is designed to take place as close to where the error occurs as possible, minimising its effect on the rest of the network and enabling error recovery to be carried out very quickly.

8. A Radiation Tolerant SpaceFibre Chip

An experimental radiation tolerant SpaceFibre interface device has been developed by University of Dundee, STAR-Dundee, Ramon Chips, ACE-IC, IHP, Airbus DS and SCI within the Very High Speed Serial Interface (VHiSSI) European Commission Framework 7 project [5]. The VHiSSI chip integrates 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 The VHiSSI device is shown in Figure 22.

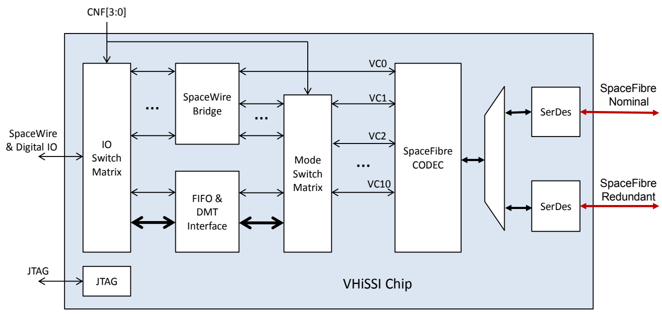

Figure 22 - VHiSSI Chip Block Diagram

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 seven port SpaceWire router which allows routing between three SpaceWire ports, three Virtual Channel (VC) buffers of the two SpaceFibre interfaces and a device configuration port. Configuration of the VHiSSI chip can be carried out over any SpaceWire interface connected to the embedded SpaceWire router or over VC0, VC1 and VC2 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. VC1 and VC2 are connected to the embedded SpaceWire router. The other VCs are connected 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 is 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 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.

The digital logic for VHiSSI was designed by STAR-Dundee Ltd. with system architectural design and project management being carried out by University of Dundee. Airbus DS provided inputs to the VHiSSI requirements. The back end design was carried out by Ramon Chips. ACE-IC designed the SerDes parts of the chip. Test vectors were prepared by STAR-Dundee and SCI with inputs from other partners. The chip was manufactured by IHP. The resulting VHiSSI chip is shown in Figure 23.

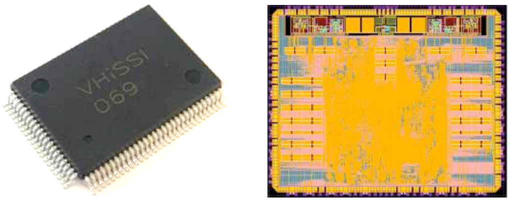

Figure 23 - VHiSSI SpaceFibre Chip

The three major modes of the VHiSSI have been tested in hardware in the laboratory. The board used for testing the SpaceWire LVTTL bridge mode of operation in shown in Figure 24.

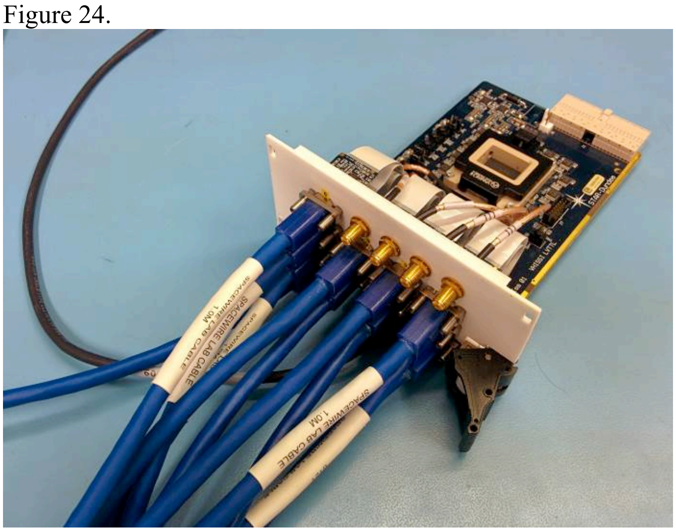

Figure 24 - VHiSSI SpaceFibre to SpaceWire LVTTL Test Board

The VHiSSI chip is housed in the centre of the board in a specially designed socket, which has a hole in the top to permit Single Event Effect radiation testing of the VHiSSI device. The 11 SpaceWire links are attached to the front panel of the test board (blue cable) and are concentrated by the VHiSSI chip over the SpaceFibre cable (black cable).

The test set up for this board is illustrated in Figure 25. Two eight-port SpaceWire routers are used to connect to all the SpaceWire ports of the VHiSSI chip. A STAR Fire unit is connected to the SpaceFibre cable and also to a host computer, via SpaceWire and a SpaceWire-USB Brick. Software on the host computer was written to exercise and test the function of the VHiSSI chip, providing a test report for each device tested.

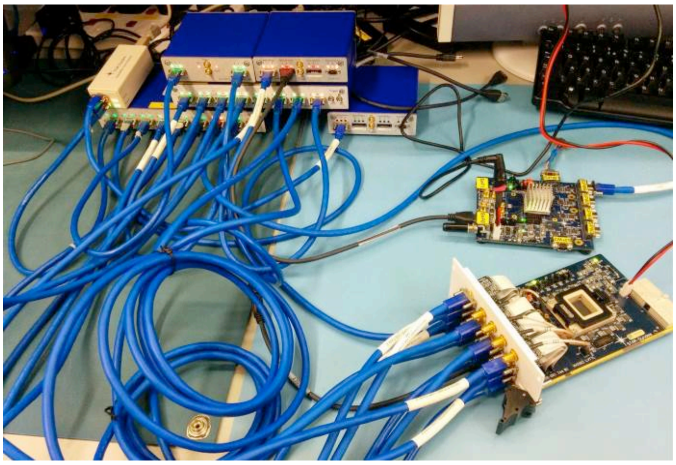

Figure 25 Test Setup for SpaceWire LVTTL Test Board

9. SpaceFibre Validation

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 SpaceFibre Demonstrator contract.

As SpaceFibre was being designed by the University of Dundee, various alternative designs were simulated to rapidly explore alternative designs and support design trade-offs.

In parallel with specifying the SpaceFibre standard the University of Dundee designed and tested the SpaceFibre IP core in VHDL. This was used to validate each revision of the SpaceFibre standard in a series of FPGA implementations.

To support the testing of SpaceFibre a suitable test platform was required, so STAR-Dundee Ltd. developed the STAR Fire unit. This device was used as a validation platform for the SpaceFibre IP core. Link analysis capability was included so that the exchange of information over the SpaceFibre interface could be recorded and analysed.

As the specification of the SpaceFibre standard developed formal simulations of the standard were carried out by St Petersburg University of Aerospace Instrumentation (SUAI), covering drafts C, D and E [6], and by Thales Alenia Space France, covering draft F3. These simulations identified many issues with the SpaceFibre standard which were then rectified.

NEC and Melco in Japan 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.

Several ESA projects are using the Dundee SpaceFibre IP core under a Beta evaluation programme. Feedback from these beta sites has been used to improve the SpaceFibre standard and the SpaceFibre VHDL IP core and related documentation.

To raise the TRL of SpaceFibre a spaceflight engineering model is being developed by Airbus Defence and Space in the frame of the ESA SpaceFibre Demonstrator project. This design uses already flight proven components (RTAX2000 and TLK2711-SP).

The VHiSSI radiation tolerant SpaceFibre interface device was developed by University of Dundee and partners within the Very High Speed Serial Interface (VHiSSI) European Commission Framework 7 project. This device has been manufactured and is currently being tested.

Axon is working on an open specification for SpaceFibre cable and connectors, which has been referred to in the current draft specification of the SpaceFibre standard. The cables and connectors have been tested using the STAR Fire unit.

10. Conclusions

SpaceFibre is a multi-Gigabit/s data link and network technology specifically designed for spaceflight applications. It is targeted primarily at spacecraft onboard payload data-handling applications. It includes built in, very efficient, quality of service and fault detection, isolation and recovery techniques, which simplify the use of SpaceFibre enormously; providing substantial system level benefits without requiring the implementation of complex performance limiting software protocols. SpaceFibre is backwards compatible with SpaceWire at the packet level allowing easy bridging between SpaceWire and SpaceFibre, so that existing SpaceWire devices can be incorporated into a SpaceFibre network and take advantages of its performance and QoS and FDIR capabilities.

SpaceFibre has been designed, reviewed and validated through analysis, simulation and hardware implementation, in a series of stages with feedback from each validation cycle feeding into the design. This has resulted in a mature well tested standard which will be released to ECSS for formal standardisation at the end of 2015. The TRL is already at TRL 5 with an implementation designed in flight proven radiation tolerant FPGA and SerDes devices. It will be raised to TRL 6 with application demonstrations in the near future. An experimental radiation tolerant SpaceFibre interface has been designed and manufactured and is currently undergoing tests.

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] 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.

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

BIOGRAPHIES

Steve Parkes *is the Director of the Space Technology Centre at the University of Dundee leading research on spacecraft on-board data-handling networks (SpaceWire & SpaceFibre), planet surface simulation, autonomous lander navigation, and digital signal and image processing for satellites. Steve wrote the ECSS-E-ST-50-12C SpaceWire standard with inputs from international spacecraft engineers, a technology that is now being used on more than 100 spacecraft. He is currently researching deterministic SpaceWire networks for integrated avionics and payload networks, SpaceFibre a multi-Gbit/s network technology for spaceflight applications, vision-based navigation for planetary landers, and FFT based spectrometers for an atmospheric chemistry instrument.*

Alberto Gonzalez Villafranca *holds a doctorate in data compression for space applications and has been connected to the space field his entire professional career. Alberto has been deeply involved in the definition and implementation of SpaceFibre since he joined STAR-Dundee Ltd. Before working with SpaceFibre he had collaborated with the Gaia mission and worked in a hardware implementation of a deterministic variant of the SpaceWire protocol at the European Space Agency.*

David McLaren *received his MEng degree in Electronic Engineering from Durham University, UK, in 2005 and EngD in System Level Integration from the University of Glasgow, UK, in 2010. He has experience in hardware and software development within industry and academia, including spacecraft simulator development, FPGA design, and research into space internetworking and on-board computing architectures. His current research at the Space Technology Centre concerns development and testing of microchips implementing the SpaceFibre protocol.*

Chris McClements *has worked as a research assistant at the University of Dundee since 2003 where he was responsible for the design of the SpaceWire 10X router which was implemented as a radiation telerant ASIC with Austrian Aerospace and Astrium (AT9710E) which is used in many ESA missions including the Bepi Columbo and Solar Orbiter missions. During this time Dr McClements was also the author and developer of the SpaceWire-B and SpaceWire RMAP VHDL IP cores which are available through the ESA IP core service. The SpaceWire IP core is the most widely used IP core in the ESA portfolio and used in many ESA missions employing FPGA and ASIC devices. He is currently working on test and development equipment for high speed serial SpaceFibre devices.*

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