SpaceWire and SpaceFibre on the Microsemi RTG4 FPGA

Steve Parkes, Chris McClements, David McLaren,
Space Technology Centre, University of Dundee, Dundee, DD1 4HN, UK
+44 1382 225613
s.m.parkes@dundee.ac.uk c.mclements@dundee.ac.uk d.y.mclaren@dundee.ac.uk:

Bassam Youssef, Mir Sayed Ali,
Microsemi, California, USA
Bassam.Youssef@microsemi.com MirSayed.Ali@microsemi.com

Albert Ferrer Florit, Alberto Gonzalez Villafranca,
STAR-Dundee Ltd., STAR House, 166 Nethergate, Dundee, DD1 4EE, UK
+44 1382 201755
Albert.Ferrer@star-dundee.com Alberto.Gonzalez@star-dundee.com

Abstract - **SpaceWire is a spacecraft on-board data-handling network which connects instruments to the mass-memory, data processors and control processors, which is already in orbit or being designed into more than 100 spacecraft. SpaceFibre is a new, multi-Gbits/s, on-board network technology, which runs over both electrical and fibre-optic cables. SpaceFibre is capable of fulfilling a wide range of spacecraft on-board communications applications because of its inbuilt quality of service (QoS) and fault detection, isolation and recovery (FDIR) capabilities.**

The Microsemi RTG4 is a new generation radiation tolerant FPGA. It has extensive logic, memory, DSP blocks, and IO capabilities and is inherently radiation tolerant, having triple mode redundancy built in. The RTG4 has a flash configuration memory built into the device. In addition the FPGA incorporates 16 SpaceWire clock-data recovery circuits and 24 multi-Gbits/s SerDes lanes to support high-speed serial protocols like SpaceFibre.

STAR-Dundee has implemented SpaceWire and SpaceFibre IP cores using the Microsemi RTG4 Development Kit. The flight proven SpaceWire IP core was initially run at over 200 Mbits/s and the SpaceFibre IP core at 2.5 Gbits/s. With a little more work it is expected to reach 300 Mbits/s and 3.125 Mbits/s respectively. The SpaceWire IP core takes around 1% of the FPGA and the SpaceFibre IP core around 3-5% depending on the number of virtual channels supported. The use of the RTG4 with the SpaceWire and SpaceFibre IP cores provides a powerful platform for future spacecraft on-board instrument control, data-handling, and data processing. Furthermore the advanced QoS and FDIR capabilities of SpaceFibre make it suitable for a wider range of spacecraft on-board applications including integrated payload data-handling and attitude and orbit control networks and launcher applications where deterministic data delivery is required.

This paper reports on the implementation and testing of the SpaceWire and SpaceFibre IP cores in the Microsemi RTG4 FPGA.

Index Terms—**SpaceWire, SpaceFibre, Network, Spacecraft On-board Data-Handling, FPGA, RTG4, Radiation Tolerant, Quality of Service, FDIR, Next Generation Interconnect.**

TABLE OF CONTENTS

1. INTRODUCTION

SpaceWire [1][2][3] is a spacecraft on-board data-handling network which connects instruments to the mass-memory, data processors and control processors. The SpaceWire standard was written by the University of Dundee with inputs from international engineers and published in 2003. Since then it has been adopted for use on many spacecraft, with over one hundred spacecraft in orbit or being designed using SpaceWire. The University of Dundee developed the ESA(b) SpaceWire IP core which is being used under licence on many ESA missions. This flight proven IP core is also available from STAR-Dundee for use on other missions.

SpaceFibre [4][5][6][7] is a new, multi-Gbits/s, on-board network technology, which runs over both electrical and fibre-optic cables. SpaceFibre is capable of fulfilling a wide range of spacecraft on-board communications applications because of its inbuilt quality of service (QoS) and fault detection, isolation and recovery (FDIR) capabilities. The innovative QoS mechanism provides concurrent bandwidth reservation, priority and scheduled QoS. Novel integrated FDIR support provides galvanic isolation, transparent recovery from transient errors, error containment and "Babbling Node" protection. 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 advantage of its performance and QoS and FDIR capabilities. SpaceFibre is now being standardised by the European Cooperation for Space Standardization (ECSS) and is expected to be published as a formal standard in 2016. It is an open standard, which is currently available in draft form from the ESA SpaceWire Working Group website [4]. 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. 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.

The Microsemi RTG4 [8] is a new generation radiation tolerant FPGA. It has extensive logic, memory, DSP blocks, and IO capabilities and is inherently radiation tolerant, having triple mode redundancy built in. The RTG4 has a flash configuration memory built into the device. In addition the FPGA incorporates 16 SpaceWire clock-data recovery circuitry and 24 multi-Gbits/s SerDes lanes to support high-speed serial protocols like SpaceFibre.

This paper reports on the implementation and testing of the SpaceWire and SpaceFibre IP cores in the Microsemi RTG4 FPGA. The Microsemi RTG4 FPGA is introduced in section 2. The facilities that the RTG4 FPGA provides to support SpaceWire are described in section 3 and to support SpaceFibre in section 4. A demonstration of SpaceWire and SpaceFibre implemented on the RTG4 FPGA is described in section 5. Lessons learnt are summarized in section 6 and conclusions are given in section 7.

2. MICROSEMI RTG4 FPGA

The RTG4 FPGA is a new radiation tolerant FPGA from Microsemi which is fully reprogrammable and which has an integrated flash memory to store configuration information. The RTG4 architecture is illustrated in Figure 1.

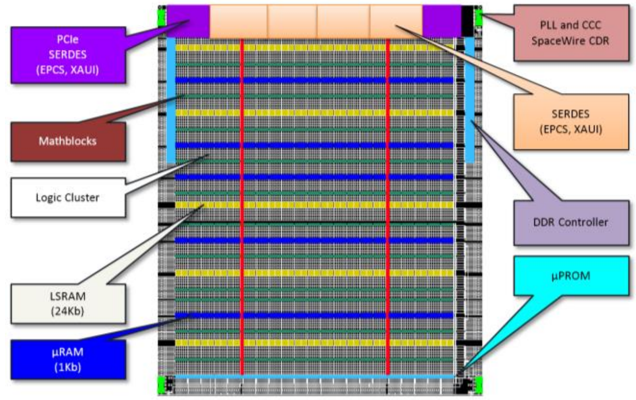

Figure 1. Microsemi RTG4 FPGA Architecture

The RTG4 contains 151,824 logic elements each comprising a 4-input LUT and flip-flop capable of operating at a clock speed of 250 MHz. There are 209 blocks of dual-port SRAM which can each be configured as 512 x 36, 1 kbit x 18, 2 kbit x 9 or 2 kbit x 12 memory blocks. The RTG4 also includes 210 three-port SRAM blocks which can each be configured as 64 x 18, 128 x 12 or 128 x 9. To access external memory there are two high-speed DDR2/DDR3 memory controllers which can operate at 333MHz and support x9, x12, x18 and x36 bus widths. The memory controllers provide optional error detection and correction (EDAC) for the external memory devices.

The RTG4 has 462 math blocks, each of which contains an 18x18-bit multiplier and a 44-bit accumulator. The math blocks can provide 250 MHz pipelined performance giving a total potential DSP performance of 230 GOPS. The math blocks are ideal for many DSP functions including filters and Fast Fourier Transforms (FFTs).

The RTG4 includes 24 high-speed (3.125 Gbits/s) serial interfaces integrated on-chip suitable for SpaceFibre. STAR-Dundee has already demonstrated SpaceFibre running over the RTG4. In addition the RTG4 has 16 SpaceWire clock and data recovery circuits, which support SpaceWire interfaces running at over 300 Mbits.

The 65nm flash process used in the RTG4 devices is intrinsically immune to configuration upsets, and the devices also feature additional radiation protection for data in flip-flops and combinatorial logic elements, embedded SRAM cells and multiply accumulate blocks. The RTG4 is designed to eliminate single-event latch-ups.

3. SUPPORT FOR SPACEWIRE ON RTG4 FPGA

SpaceWire uses Data-Strobe (DS) encoding. This is a coding scheme which encodes the transmission clock with the data into Data and Strobe so that the clock can be recovered by simply XORing the Data and Strobe lines together. The data values are transmitted directly and the strobe signal changes state whenever the data remains constant from one data bit interval to the next. This coding scheme is illustrated in Figure 2. The reason for using DS encoding is to improve the skew tolerance to almost 1-bit time, compared to 0.5 bit time for simple data and clock encoding.

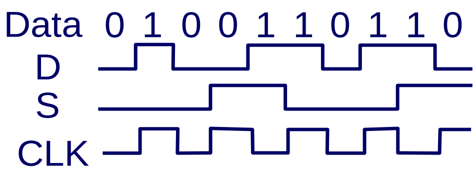

Figure 2 Data-Strobe (DS) Encoding

A SpaceWire link comprises two pairs of LVDS differential signals, one pair transmitting the D and S signals in one direction and the other pair transmitting D and S in the opposite direction. That is a total of eight wires for each bi-directional link.

The RTG4 FPGA has SpaceWire receive clock recovery blocks integrated in the FPGA fabric, to simplify the design of SpaceWire interfaces.

RTG4 SpaceWire Clock Recovery Block Overview

The SpaceWire RX clock recovery blocks reside in the clock conditioning circuitry (CCCs) parts of the FPGA fabric, which are located in each corner of the FPGA, see Figure 1.

There are two RX clock recovery blocks in each CCC block, see Figure 3. Each RX clock recovery block has multiplexing logic to select the two inputs for data and strobe. The data input is also passed to FPGA fabric so that it can be used as the input to the SpaceWire CODEC logic, which is implemented by the user in the FPGA fabric.

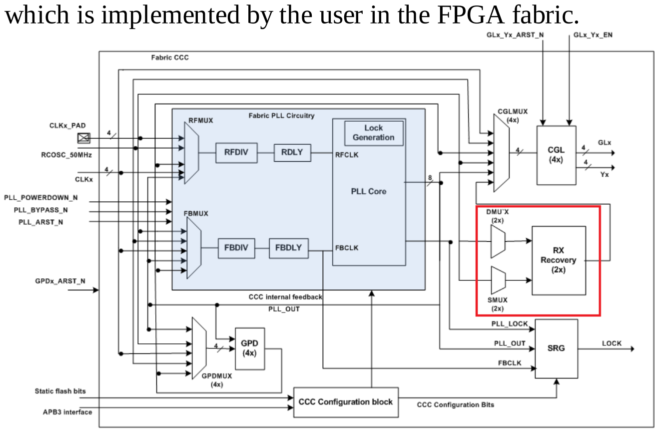

Figure 3. RTG4 SpaceWire RX clock recovery block in CCC

Figure 4 shows the implementation of the RX clock recovery block in the RTG4. The Data and Strobe inputs come from dedicated input pads and can be configured as either single ended (LVTTL) or differential (LVDS). The Data and Strobe signals are passed into an XOR gate which reproduces the clock signal. The clock signal is then passed through a de-glitching circuit to prevent any unwanted narrow clock pulse at the RX clock output. The de-glitching circuit has a delay setting that is used to add optional filtering to filter either SET or system-level glitches. Enabling the delay is done globally through a configuration option in the Libero SoC software. The generated RX clock is radiation hardened and drives the hardened global clock network.

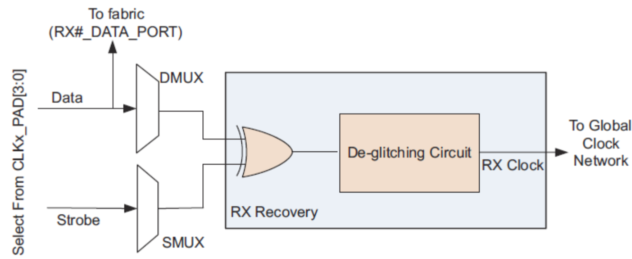

Figure 4. RTG4 SpaceWire RX clock recovery block

The RTG4 device has 16 SpaceWire clock recovery blocks. Note that the current RTG4 ES silicon only has 12 sets of SpaceWire data and strobe input pins, however, the production silicon will have 16 sets of SpaceWire data and strobe input pins.

Configuring the SpaceWire Clock Recovery Block

The RX clock recovery block is accessed by using the Clock Conditioning Circuit (CCC) configurator in Libero SoC to configure the RX clock recovery block. The RX clock recovery block generates the RX clock that can drive any one of the global outputs of the CCC. The steps required to configure the SpaceWire clock recovery block in the CCC configurator, shown in Figure 5, are listed below:

1. Select the desired output clocks from the four different global clocks (GL0, GL1, GL2, and GL3). In addition, it is possible to select the four core clocks (Y0, Y1, Y2, and Y3).

2. For each selected output clock, select the desired reference (input) clock from which the output will be derived. It can be either RX0 Clock Recovery or RX1 Clock Recovery.

3. Select the Data pad: CLK_PAD1 or CLK_PAD3. The Strobe pad will be automatically selected.

4. Generate the RTG4 CCC configurator.

The dedicated input pad for data and strobe can also be configured as either single-ended or differential. For the differential option, picking one pad of the differential pair automatically selects the other pad of the differential pair as well.

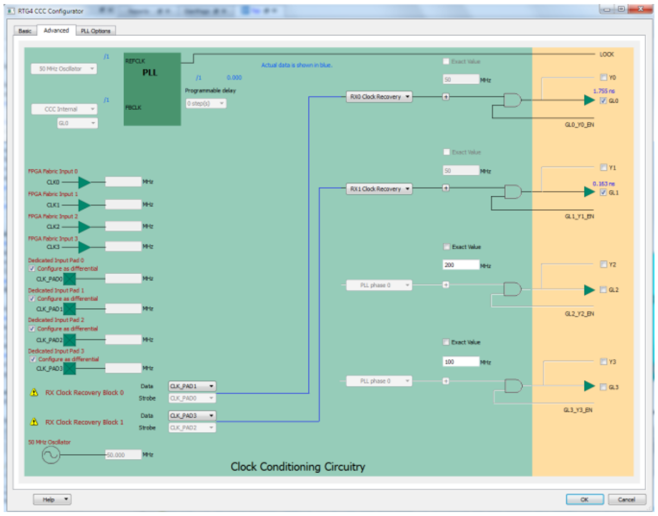

Figure 5. SpaceWire RX clock recovery block selection in the CCC configurator RTG4 data recovery block Implementation

Figure 6 shows a basic SpaceWire data recovery block implementation in an RTG4 device. The hardened RX clock recovery block is used to recover the clock and a double data-rate input register (DDR_IN macro) is used to capture the data on both the rising and falling edges of the RX clock. The recovered clock and recovered data can then be used as inputs to the rest of the SpaceWire CODEC receiver logic. The SpaceWire receive clock recovery circuit greatly simplifies the place and route of SpaceWire CODECs and enables SpaceWire interfaces with higher performance.

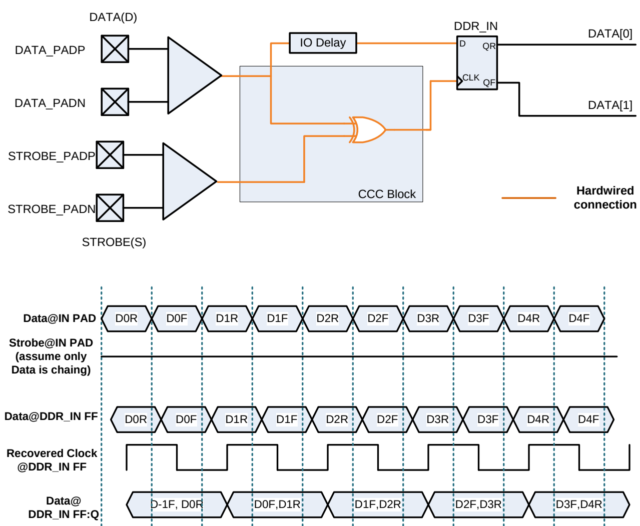

Figure 6. RTG4 SpaceWire Clock and Data Recovery block and waveform

4. SUPPORT FOR SPACEFIBRE ON RTG4 FPGA

To implement SpaceFibre a high-speed serialiser/de-serialiser (SerDes) is required, which takes relatively slow, parallel data and serialises it for transmission at high-speed and de-serialises the received serial data to recover the parallel data stream. In the past external SerDes devices have been used with FPGAs, such as the RTAX2000 from Microsemi, to provide a high-speed serial interface. STAR-Dundee has implemented SpaceFibre on such an FPGA using a separate SerDes chip, the TLK2711-SP from Texas Instruments.

The RTG4 FPGA includes radiation tolerant SerDes on-chip avoiding the need for external SerDes devices. These integrated SerDes make the RTG4 ideal for the implementation of spaceflight data-handling and processing sub-systems with multi-Gbits/s SpaceFibre interfaces.

In this section the support that the RTG4 provides for SpaceFibre, in the way of high-speed SerDes, is described.

RTG4 Serialiser/De-serialiser Module

The RTG4 FPGA contains 24 general purpose, high-speed serial/de-serialiser (SerDes) lanes. The SerDes lanes are within the Serial Sub-System of the RTG4, see Figure 1. The SerDes are formed into groups of four, which are known as SerDes modules. Each SerDes is also called a physical media attachment (PMA), which is the name used in PCIe. Eight of the 24 SerDes interfaces can be used to implement PCIe.

The RTG4 SerDes module is illustrated in Figure 7.

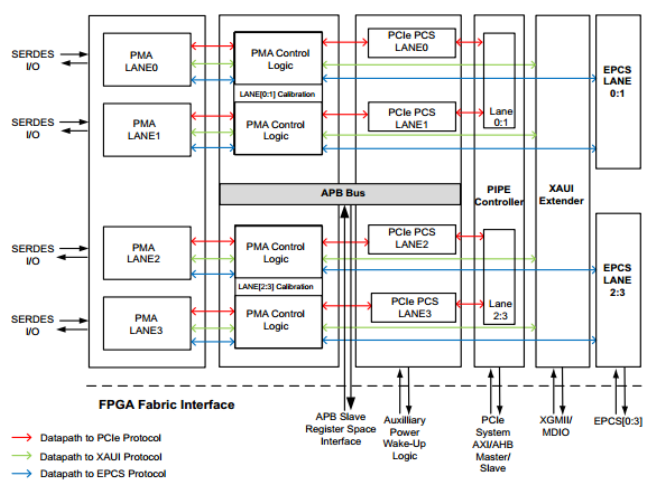

Figure 7. RTG4 SerDes Module Block Diagram

Each of the SerDes blocks includes:

- **SerDes Module**: The SerDes module implements four high-speed SerDes lanes with associated transmit driver and receiver. Each SerDes is shown as a PMA Lane in Figure 7. The PMA Control Logic is configured to pass data to/from the EPSC interfaces.

- **SerDes Module System Registers**: The SerDes module system registers control the SerDes module for single protocol or multi-protocol support implementation. These registers can be accessed through a 32-bit APB interface, and the default values of these registers are configured using Libero® System-on-Chip (SoC) software. Each SerDes module is initially configured at power-up with parameters determined during the FPGA design flow using the Libero SoC software. The SerDes module configuration can subsequently be changed by writing to the related control registers through the advanced peripheral bus (APB) interface.

- **EPCS block**: The External Physical Coding Sub-Layer (EPCS) block provides a low-level interface to the SerDes to support the implementation of various serial communications protocols in the FPGA fabric.

- **PCIe System**: The PCIe System block implements a x1, x2, and x4 lane PCIe endpoint interface.

- **XAUI Extender**: The XAUI block is an XGMII extender to support the XAUI protocol through a FPGA IP MAC core in the FPGA fabric.

Each SerDes module interfaces with fabric, program control, and four duplex SerDes differential I/O pads.

SerDes Operation

The block diagram of an RTG4 SerDes is provided in Figure 8.

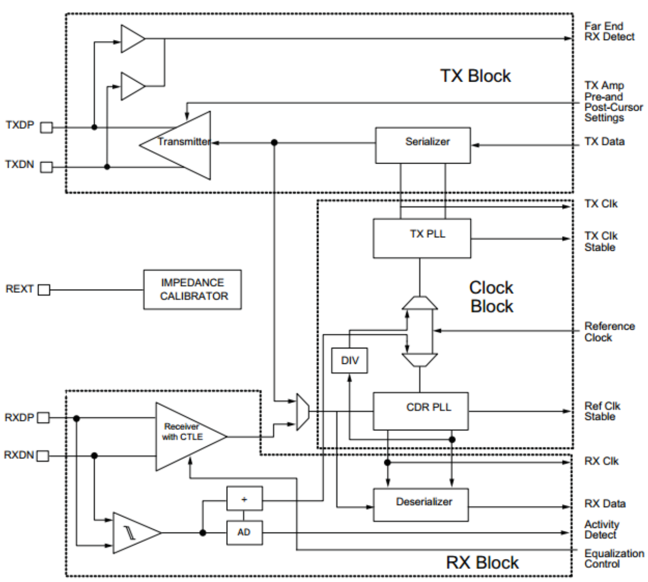

Figure 8. SerDes Detailed Block Diagram

The data to be transmitted by the SerDes (TX Data) is passed in parallel words to the serialiser. The transmit clock for the serialiser is generated by a transmit phase-locked loop (PLL) which takes a reference clock and multiplies it up to give the required serial bit rate. The reference clock is at the same rate as the parallel words going into the serialiser. The serialiser takes each parallel word and converts it into a serial bit stream at the appropriate bit rate and passes it to a line transmitter which provides a current mode logic (CML) serial output.

The received serial data comes into the CML receiver and is passed to a deserialiser. A clock data recovery (CDR) circuit generates a clock signal from the received bit stream. This is then used to clock the deserialiser, which converts the serial input stream into parallel words. A signal detection circuit is used to detect when a receive signal is present on the input of the receiver. The inverse of this signal is used to provide the Loss of Signal indication required for SpaceFibre.

EPCS Interface

The EPCS interface provides a means of accessing the capabilities of each SerDes module from the FPGA fabric via an up to 20-bit parallel interface. It also provides a means of accessing the SerDes control and status registers through the APB interface.

Radiation Hardening

Within the high-speed SERDES block, radiation hardening techniques are applied to the APB, REFCLK, and SPLL-used with XAUI and PCIE blocks. This is done with various types of mitigation techniques including TMR logic triplication and self-correcting latches as well as added digital delay buffers. The asynchronous resets are hardened using glitch-suppression filters and by using special buffers for reset distribution in all areas of the circuit. Two outputs of the clocks generated out of each SERDES block can be routed directly onto a global clock network which is radiation hardened.

5. SPACEWIRE AND SPACEFIBRE DEMONSTRATION SYSTEM

To explore the implementation of SpaceWire and SpaceFibre in the RTG4 FPGA, a demonstration/evaluation design was developed to run on the RTG4 development board. A block diagram of this demonstration design is shown in Figure 9

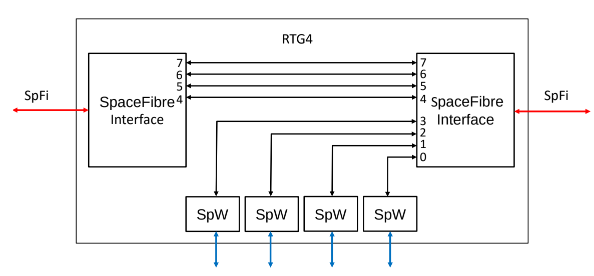

Figure 9. Functional block diagram showing interconnection between SpaceFibre and SpaceWire interfaces in an RTG4 FPGA

There are two SpaceFibre interfaces: one with eight virtual channels and the other with four. Four SpaceWire interfaces are connected to virtual channels 0 to 3 on one SpaceFibre interface and virtual channels 4 to 7 are interconnected between the two SpaceFibre interfaces.

The complete demonstration system has two of the RTG4 boards connected together as shown in Figure 10.

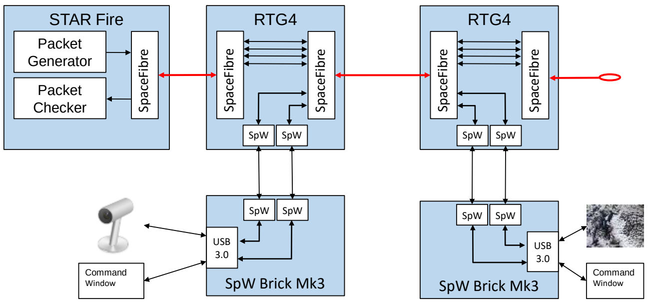

Figure 10. Block diagram of complete SpaceFibre and SpaceWire RTG4 demonstration system

In addition to the RTG4 board there is a pair of STAR-Dundee SpaceWire Brick Mk3 devices and a STAR Fire unit. The SpaceWire Brick Mk3 provides an interface between USB 3.0 and two SpaceWire interfaces. The STAR Fire unit provides several functions: SpaceWire to SpaceFibre bridge, SpaceFibre packet generator and checker, and SpaceFibre link analyser. The STAR Fire unit is connected to one SpaceFibre interface of an RTG4 device. The other SpaceFibre interface of this RTG4 is connected to the second RTG4 and the second SpaceFibre interface of this RTG4 board has a loopback cable fitted. Two SpaceWire Brick Mk3 devices (USB to SpaceWire interfaces) are connected to the SpaceWire ports on the two RTG4 boards. One SpaceWire Brick Mk3 is connected to a PC with a web-camera and the other Brick Mk3 is connected to another PC acting as a display.

The packet generator in the STAR Fire device generates SpaceFibre packets at up to the full data rate of the SpaceFibre link. The generated packets travel over one of the SpaceFibre virtual channels to the first RTG4 device where they are received, transferred to the other SpaceFibre interface and sent out of the same virtual channel of that SpaceFibre interface. The SpaceFibre packets travel across the SpaceFibre link to the second RTG4 board, across the FPGA, out of its second SpaceFibre interface to the loopback cable. The SpaceFibre packets then take the path back to the STAR Fire unit where they are checked for errors by the packet checker.

The PC fitted with the webcam grabs images, puts each image into a SpaceWire packet and sends the packets over the SpaceWire Brick Mk3 unit to the RTG4 device. The SpaceWire packets are received by the RTG4 and passed to a virtual channel of the SpaceFibre interface. Since SpaceWire and SpaceFibre packets have identical format, this is very simple to do. The SpaceWire packets travel over SpaceFibre to the second RTG4 and then out of the SpaceWire interface to the second Brick Mk3. They are passed up to the host computer where they are displayed.

The complete demonstration system is shown in Figure 11.

Figure 11. Photograph of SpaceFibre and SpaceWire RTG4 demonstration system

The SpaceFibre and SpaceWire interfaces operated correctly on the RTG4 and various QoS and FDIR capabilities have been demonstrated. The interconnection of two RTG4 devices via SpaceFibre has been demonstrated and the interconnection of a Xilinx FPGA (in the STAR Fire unit) to the RTG4 via SpaceFibre has also been demonstrated. The flight proven SpaceWire IP core was initially run at over 200 Mbits/s and the SpaceFibre IP core at 2.5 Gbits/s. With a little more work it is expected to reach 300 Mbits/s and 3.125 Mbits/s respectively. The FPGA resources used to implement a SpaceWire and SpaceFibre IP core in the demonstration design are listed in Table 1. The number of flip-flops, look-up tables (LUTs) and RAM blocks used are detailed in the table, along with the percentage of those elements of the complete RTG4 device. In addition each SpaceFibre interface uses one of the inbuilt radiation tolerant SerDes elements of the RTG4.

Table 1. RTG4 Resource Usage

	Flip Flops	**LUTs**	**RAM**
SpaceWire	380 0.25%	520 0.34%	2xRAM64x18 0.95%
SpaceFibre 4 VCs	3967 2.61%	7219 4.75%	18xRAM1kx18 8.61%

The demonstration design was not optimized in its use of RAM blocks. Further design work has shown that the number of RAM blocks can be reduced significantly resulting in a SpaceFibre interfacing using around 3-5% of an RTG4 FPGA depending on the number of virtual channels.

6. LESSONS LEARNT

It is straightforward to use the SpaceWire and SpaceFibre IP cores in the RTG4 FPGA. The complete demonstration design took about three person-months to implement, starting from generic SpaceWire and SpaceFibre IP cores, customising them for the RTG4, and implementing and testing the overall design. The most difficult part of the work was the appropriate configuration of the RTG4 SerDes. This has also been the case when implementing the SpaceFibre interface in other FPGAs.

SpaceFibre provides high-speed communications and is very easy to integrate with SpaceWire. The connection of the four SpaceWire interfaces into four virtual channels of one of the SpaceFibre interfaces was simple, because SpaceWire and SpaceFibre have the same packet format.

This will be added in the first quarter of 2016. The SpaceFibre IP core was developed in parallel with the development and specification of the SpaceFibre standard. It has been used to prove and validate the various concepts incorporated into SpaceFibre including the quality of service and fault detection, isolation and recovery capabilities. The main limitation of the SpaceFibre IP core used in the demonstration design is that it does not include multi-lane capability.

The RTG4 FPGA was selected because it is radiation tolerant, including the necessary triple mode redundancy (TMR) in the fabric of the FPGA, and it has the essential SerDes embedded on-chip. SpaceFibre has also been implemented in the Microsemi RTAX FPGA with an external SerDes and in several Xilinx FPGAs which have embedded SerDes including the Spartan 6, Virtex 4, Virtex 5, Virtex 6 and Kintex 7.

SpaceFibre is a protocol that can run in various FPGA devices. To achieve TRL 6 it is necessary for the FPGA device to be flight qualified and for the SpaceFibre design to be tested in a representative implementation. The RTG4 is currently being tested and qualified for space applications. Several representative implementations for the SpaceFibre IP core are currently being designed by STAR-Dundee and other organisations.

SpaceFibre provides a small footprint, high performance, and high capability serial communication interface for a spaceflight FPGA. An RTG4 FPGA can implement digital signal and image processing units, control processors, mass memory controllers, telemetry controllers and other on-board functions and include SpaceFibre on the same chip for high-speed communication. SpaceFibre is backwards compatible with SpaceWire at the packet level making bridging between SpaceWire and SpaceFibre trivial. SpaceWire equipment can be readily integrated in a SpaceWire network.

7. CONCLUSIONS

STAR-Dundee has implemented SpaceWire and SpaceFibre IP cores on the Microsemi RTG4 FPGA using the RTG4 Development Kit. The flight proven SpaceWire IP core was initially run at over 200 Mbits/s and the SpaceFibre IP core at 2.5 Gbits/s. With a little more work it is expected to reach 300 Mbits/s and 3.125 Mbits/s respectively. The SpaceWire IP core takes around 1% of the FPGA and the SpaceFibre IP core around 3-10% depending on the number of virtual channels supported. The use of the RTG4 with the SpaceWire and SpaceFibre IP cores provides a powerful platform for future spacecraft on-board instrument control, data-handling, and data processing. Furthermore the advanced QoS and FDIR capabilities of SpaceFibre make it suitable for a wider range of spacecraft on-board applications including integrated payload data-handling and attitude and orbit control networks and launcher applications where deterministic data delivery is required.

ACKNOWLEDGMENT

The research leading to these results has received funding from 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 and from the UK Space Agency via the Centre for Earth Observation Instrumentation and Space Technology (CEOI-ST).

REFERENCES

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

[2] S. Parkes, P. Armbruster and M. Suess, "SpaceWire On-Board Data-Handing Network", ESA Bulletin, Volume 145, pp 34-45, February 2011.

[3] S. Parkes, "SpaceWire Users Guide", ISBN: 978-0-9573408-0-0, STAR-Dundee, 2012, available from https://www.star-dundee.com/sites/default/files/SpaceWire%20User%27s%20Guide.pdf (last visited 23rd Sept 2015).

[4] S. Parkes, A. Ferrer and A. Gonzalez, "SpaceFibre Standard", Draft H1 September 2015, available from http://space-env.esa.int/indico/confLogin.py?confId=32

[5] S. Parkes et al, "SpaceFibre: Multi-Gigabit/s Interconnect for Spacecraft On-board Data Handling", IEEE Aerospace Conference, Big Sky, Montana, 2015.

[6] 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.

[7] S. Parkes, C. McClements, A. Ferrer and A. Gonzalez, "SpaceFibre: Multi-Gbits/s Network for Spaceflight Applications", paper no. IAC-13-B2.3, 64th International Astronautical Congress, Beijing 2013.

[8] DS0131: RTG4 FPGA Datasheet

[9] SpaceWire Pin Mapping from RTG4 ES/MS Silicon to PROTO/Flight Silicon

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.*

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

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 on a hardware implementation of a deterministic variant of the SpaceWire protocol at the European Space Agency.*

Bassam Youssef *holds a Bachelor's degree in electronics engineering from Georgia Tech and a Master of Science degree in computer engineering from San Jose State. Currently Bassam is a Senior Staff Application Engineer at Microsemi Systems on Chip Products Group. He brings nearly 13 years of experience in the areas of FPGA software, design, verification and implementation. He has worked on reference designs for Hi-rel, Embedded and low power. He has expertise in Microsemi's Rad-Tolerant FPGAs and he is a lead engineer on supporting RTG4.*

Mir Sayed Ali *is the Principal Corporate Applications Engineer for Microsemi's Systems on Chip Products Group. He brings nearly 15 years of experience in the areas of FPGA design, verification and implementation. He has worked on reference design for Hi-rel, Embedded, low power, Industrial, communication, DSP market. He has expertise in Microsemi's Rad-Tolerant FPGAs and Microsemi space applications IP such as 1553, PCIe/PCI and DSP. Mir holds a Master's degree in Electrical Engineering from University of Oklahoma, Norman, USA.*

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 tolerant 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.*