Efficient High Data Rate Networking Using Remote Direct Memory Access Over SpaceFibre

Dave Gibson
STAR-Dundee
Dundee, Scotland, UK
david.gibson@star-dundee.com

Andrew MacLennan
STAR-Dundee
Dundee, Scotland, UK
andrew.maclennan@star-dundee.com

Stuart Mills
STAR-Dundee
Dundee, Scotland, UK
stuart.mills@star-dundee.com

Steve Parkes
STAR-Dundee
Dundee, Scotland, UK
steve.parkes@star-dundee.com

Abstract— **SpaceFibre [1] provides multi Gbit/s on-board networking for spaceflight applications. It has in-built quality-of-service (QoS); fault detection, isolation, and recovery (FDIR); and can run over electrical or fibre-optic cables. Using multiple lanes, it supports very high data rates. For example, with four lanes running at a lane signalling rate of 7.8125 Gbit/s, an overall link signalling rate of 31.25 Gbit/s is achieved. After adjusting for 8b/10b encoding and protocol overheads, this results in an approximate unidirectional data rate of 23.8 Gbit/s and bidirectional data rate of 23.1 Gbit/s.**

To support such high data rates in embedded systems with minimal processor overhead, this paper presents a SpaceFibre Endpoint that uses a Remote Direct Memory Access (RDMA) approach to provide zero-copy transferring of data between user-space applications over a SpaceFibre network.

This paper first provides an overview of RDMA over SpaceFibre, then presents performance test results, and lastly describes an image transfer demonstration showing RDMA over SpaceFibre used in a real application.

Keywords— SpaceFibre, On-board Data Handling, Networks, Remote Direct Memory Access, Embedded Systems

I. Introduction

The initial prototype of the SpaceFibre Endpoint described in this paper was researched and developed as part of the Hi-SIDE project [2], which was a European Union Horizon 2020 project that ended in 2022. The objective of the Hi-SIDE project was to develop and demonstrate technologies for high-speed on-board data-handling interconnected via a SpaceFibre network [3][4].

As part of the Hi-SIDE project, a SpaceFibre Endpoint was prototyped to provide a high-performance direct interface between an embedded processing system and a SpaceFibre network. With regards to performance, the objective was to maximise link utilisation while minimising processor overhead. To achieve this objective, an RDMA-based architecture was selected. The initial prototype was implemented in the Field Programmable Gate Array (FPGA) of a STAR-Dundee STAR-Ultra Peripheral Component Interconnect Express (PCIe) board [5], with a user-space Application Programming Interface (API) and kernel-space driver developed for Linux.

Following on from the end of the Hi-SIDE project, development of the SpaceFibre Endpoint continued in several ways. Firstly, an Advanced eXtensible Interface 4 (AXI4) version of the SpaceFibre Endpoint was developed for implementation in the FPGA of platforms such as the Xilinx Zynq UltraScale+ series [6]. Secondly, a target-only version of the SpaceFibre Endpoint was developed for both PCIe and AXI4 platforms to allow physical memory to be made available using RDMA operations without requiring a processor on the target. Thirdly, based on initial performance results using the prototype, the hardware and software were optimised to improve performance further. Lastly, a demonstration system was developed with an image transfer application used to read images from a target endpoint over SpaceFibre into memory in an initiator endpoint and display them at high resolution using a DisplayPort interface with minimal processor overhead.

The following sections provide an overview of RDMA over SpaceFibre, present performance results for various test cases, and describe the image transfer demonstration.

II. RDMA Over SpaceFibre

In traditional networking software, data is passed from user-space applications to the network interface using a software stack often involving copying of data between the user-space application and kernel-space driver, initiating DMA operations, and other processor work.

RDMA is an approach to networking that removes much of this processor work by allowing network interfaces to access user-space application memory directly, providing low processor overhead and zero-copy transferring of data directly to or from user-space memory in remote processor systems.

The RDMA over SpaceFibre system described in this paper is based on existing RDMA technologies such as the Virtual Interface Architecture (VIA) [7] which provides an abstract model for RDMA, and implementations based on VIA such as RDMA Over Converged Ethernet (RoCE), and InfiniBand, which are both described in the InfiniBand Architecture Specification [8].

A. Operations

An RDMA operation in a SpaceFibre network is a transaction between an initiator endpoint and a target endpoint that performs one of the following:

- RDMA Read: reads data directly from virtual or physical memory in the target endpoint to user memory in the initiator endpoint:
    1. The software registers an RDMA memory region with the initiator endpoint to store data read from a target endpoint’s memory region.

2. The software submits an RDMA Read work request to the initiator endpoint.

3. The initiator endpoint converts the request into an RDMA Read request packet and sends it to the target endpoint over SpaceFibre.

4. The target endpoint receives the RDMA Read request packet, retrieves the requested data from the relevant memory region, then sends an RDMA Read response packet containing the data back to the initiator endpoint.

5. The initiator endpoint receives the RDMA Read response packet, writes the received data to the relevant memory region, then notifies the software of its completion.

- RDMA Write: writes data directly from user memory in the initiator endpoint to virtual or physical memory in the target endpoint:

  1. The software registers an RDMA memory region with the initiator endpoint containing data to write to a target endpoint's memory region.

  2. The software submits an RDMA Write work request to the initiator endpoint.

  3. The initiator endpoint retrieves the data from the memory region, converts the request into an RDMA Write request packet containing the data, and sends it to the target endpoint over SpaceFibre.

  4. The target endpoint receives the RDMA Write request packet, writes the received data to the relevant memory region, then (optionally) sends an RDMA Write response packet back to the initiator endpoint.

  5. The initiator endpoint receives the RDMA Write response packet then notifies the software of its completion.

In addition to RDMA operations, the SpaceFibre Endpoint also supports traditional sending and receiving of messages using the same underlying zero-copy approach.

B. Layers

On the initiator side, the following layers are involved:

- Application (user-space).
- SpaceFibre RDMA API (user-space).
- SpaceFibre RDMA driver (kernel-space).
- SpaceFibre Endpoint (FPGA).
- SpaceFibre Interface (FPGA).

On the target side, the endpoint may be a full endpoint, i.e., it is used with an operating system and a user-space application, or it may be a target-only endpoint, providing one or more physical memory regions. For a target-only endpoint, an operating system and user-space application are not required but may optionally be used to receive notifications of RDMA operations. Therefore, on the target side, the following layers are involved:

- Full endpoint:

  1. Application (user-space).
  2. SpaceFibre RDMA API (user-space).
  3. SpaceFibre RDMA driver (kernel-space).
  4. SpaceFibre Endpoint (FPGA).
  5. SpaceFibre Interface (FPGA).

- Target-only endpoint:

  1. SpaceFibre Endpoint (FPGA).
  2. SpaceFibre Interface (FPGA).

As far as possible, functionality is implemented in the user-space API instead of the kernel-space driver to minimise the overhead of context switching. Therefore, the kernel-space driver is mainly used for device enumeration, initialisation, and resource management.

C. Remote Memory Access Protocol

The underlying packet formats used for RDMA over SpaceFibre request and response packets are similar to the Remote Memory Access Protocol [9] with differences including:

- A 64-bit address space compared to 40-bits.
- A maximum operation size of 4 Gigabytes (GB) compared to 16 Megabytes (MB).
- Keys are used to identify remote memory regions.
- Support for sending and receiving of messages as well as executing memory operations.

Additionally, RDMA over SpaceFibre is a full network stack including the user-space API, kernel-space driver, SpaceFibre endpoint, SpaceFibre interface, and underlying protocol, all designed together to enable very high-performance with minimal processor overhead.

III. PERFORMANCE RESULTS

Performance results were gathered using a test system consisting of the following components:

- A full SpaceFibre Endpoint implemented in a Xilinx ZCU102 Zynq UltraScale+ board [10].
- A target-only SpaceFibre Endpoint implemented in a STAR-Ultra PCIe board, providing 8 GB of Double Data-Rate 3 (DDR3) memory.
- A quad-lane SpaceFibre interface [11] with a lane signalling rate of 7.8125 Gbit/s (31.25 Gbit/s link rate) connected between the two boards using an adaptor cable to connect the four Small Form-Factor Pluggable (SFP) interfaces on the ZCU102 to the Quad SFP (QSFP) interface on the STAR-Ultra PCIe board.
- The SpaceFibre RDMA user-space API, kernel-space platform driver, and user-space performance test application running on top of PetaLinux 2022.1 on the ZCU102's ARM Cortex-A53 Central Processing Unit (CPU).

A photograph of the performance test system is provided in Fig. 1.

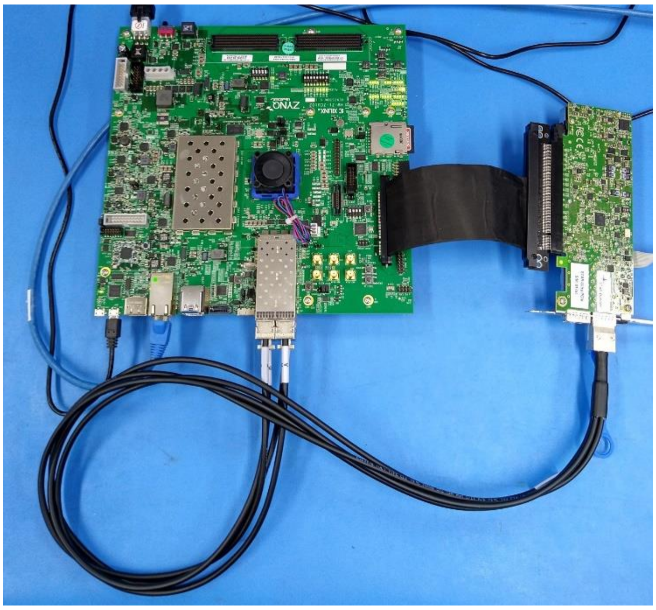

Fig. 1. SpaceFibre Endpoint Performance Test System

In Fig. 1, the ZCU102 board is on the left, and the STAR-Ultra PCIe board is on the right. The 4xSFP-to-QSFP adaptor cable connects the two boards. The PCIe adaptor cable provides power to the STAR-Ultra PCIe board but is not used for any transferring of data.

A. Single and Grouped Operations

As the RDMA approach provides zero-copy transferring of data, the processor overhead of executing a single transaction is consistent between different data lengths. The work performed by the processor, for a single operation, is a copy of the descriptor to the work queue, a wait for the completion via interrupt or polling, and a retrieval of the completion from the completion queue.

Operations can be grouped together to move the interrupt or polling overhead to a per-group overhead instead of per-operation, reducing processor overhead further.

Therefore, the best performance can be achieved using single large operations or groups of smaller operations. The following section provides performance results for single and groups of RDMA read and write operations of varying sizes.

B. RDMA Read and Write Performance

Each test case, varying from 4 KB to 512 KB data length and single or groups of 8 to 128 operations, was executed for 10 iterations with 10 seconds per iteration. Each test case is listed as the data length and group size, and each result is listed as the average followed by the worst-case in parentheses.

The single RDMA write results are listed in Table I.

TABLE I. SINGLE RDMA WRITE RESULTS

Test Case	Data Rate (Gbit/s)	CPU Utilisation (%)
4 KB x 1	2.60 (2.59)	17.64 (18.10)
8 KB x 1	4.20 (4.19)	15.26 (15.94)
16 KB x 1	7.32 (7.32)	14.87 (16.39)
32 KB x 1	11.35 (11.34)	10.62 (11.27)
64 KB x 1	15.61 (15.59)	7.91 (8.29)
128 KB x 1	19.21 (19.19)	7.49 (8.59)
256 KB x 1	21.47 (21.47)	2.56 (2.67)

The grouped RDMA write results are listed in Table II.

TABLE II. GROUPED RDMA WRITE RESULTS

Test Case	Data Rate (Gbit/s)	CPU Utilisation (%)
4 KB x 128	17.93 (17.93)	7.04 (7.34)
8 KB x 128	20.47 (20.47)	3.67 (3.75)
16 KB x 128	22.03 (22.03)	2.82 (3.00)
32 KB x 64	22.77 (22.77)	1.95 (2.21)
64 KB x 32	23.16 (23.15)	0.74 (0.84)
128 KB x 16	23.35 (23.35)	0.51 (0.62)
256 KB x 8	23.45 (23.44)	0.48 (0.58)

The single RDMA read results are listed in Table III.

TABLE III. SINGLE RDMA READ RESULTS

Test Case	Data Rate (Gbit/s)	CPU Utilisation (%)
4 KB x 1	2.01 (2.00)	15.72 (16.73)
8 KB x 1	3.71 (3.70)	14.46 (16.48)
16 KB x 1	6.41 (6.40)	13.07 (13.77)
32 KB x 1	10.12 (10.11)	10.37 (10.97)
64 KB x 1	14.21 (14.21)	3.51 (5.51)
128 KB x 1	17.81 (17.80)	4.22 (4.41)
256 KB x 1	20.40 (20.39)	2.23 (2.40)

The grouped RDMA read results are listed in Table IV.

TABLE IV. GROUPED RDMA READ RESULTS

Test Case	Data Rate (Gbit/s)	CPU Utilisation (%)
4 KB x 128	17.73 (17.73)	5.05 (5.11)
8 KB x 128	20.33 (20.33)	3.49 (3.55)
16 KB x 128	21.96 (21.96)	1.63 (1.69)
32 KB x 64	22.64 (22.64)	0.47 (0.63)
64 KB x 32	23.01 (23.01)	0.71 (0.79)
128 KB x 16	23.19 (23.18)	0.51 (0.59)
256 KB x 8	23.28 (23.28)	0.42 (0.47)

As listed in Tables II and IV, for groups of operations with a length of 64 KB or above, a data rate in excess of 23 Gbit/s can be achieved with 1% CPU utilisation or less. Compared to the maximum unidirectional data rate possible on a 31.25 Gbit/s link, which is approximately 23.8 Gbit/s, the link utilisation is over 95% calculated using the worst-case results from these test cases.

IV. IMAGE TRANSFER DEMONSTRATION

The following sections describe a demonstration of RDMA over SpaceFibre in a system used to read images from a target endpoint, over SpaceFibre, into memory in an initiator endpoint and output them on a DisplayPort interface with low processor overhead.

A. Hardware Setup

The image transfer demonstration uses a similar hardware setup as the test system used to gather performance results, shown in Fig. 1. The only other addition is inclusion of the DisplayPort interface in the design, and a DisplayPort cable connected between the ZCU102 and a 2560x1440 resolution 60 Hz monitor.

A photograph of the image transfer demonstration hardware while running is provided in Fig. 2.

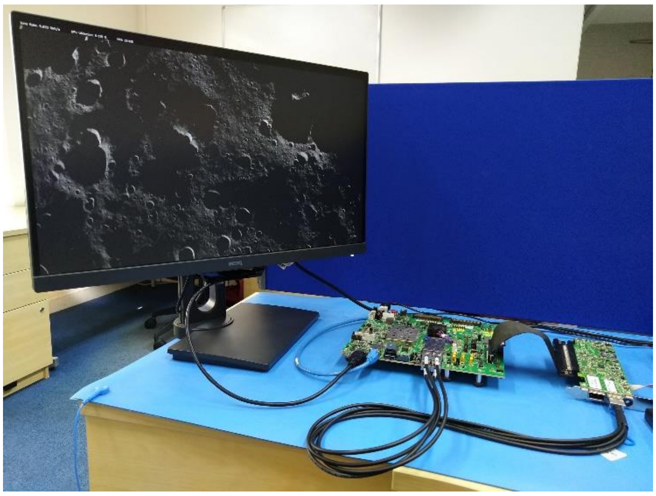

Fig. 2. SpaceFibre Endpoint Image Transfer Demonstration

The ZCU102 board's DisplayPort interface supports a maximum of 2560x1440 (1440p) resolution at 60 Hz, or 3840x2160 (4K) resolution at 30 Hz. With a 32-bit pixel depth, this results in a maximum data rate of approximately 7 Gbit/s at 1440p resolution or 7.9 Gbit/s at 4K resolution.

In this case, the demonstration uses 1440p resolution images with 32-bit pixel depth, resulting in 14,745,600 bytes per image. The STAR-Ultra PCIe board has 8 GB of DDR3, allowing it to store 582 images, or approximately 9.7 seconds of video when displayed at 60 frames per second.

B. Software Setup

The image transfer demonstration uses the same user-space API and kernel-space platform driver as the performance tests. However, the API and driver were extended to support shared contiguous buffers between multiple devices using the Linux kernel's framework for exporting and importing device buffers. This allows contiguous buffers allocated by one driver to be shared with another and mapped into another device's address space. In this case, the graphics driver allocates and exports contiguous frame buffers which are imported by the RDMA driver and made available as memory regions accessible using RDMA read or write operations.

C. Demonstration Procedure

A block diagram of the image transfer demonstration, including software and hardware, is provided in Fig. 3.

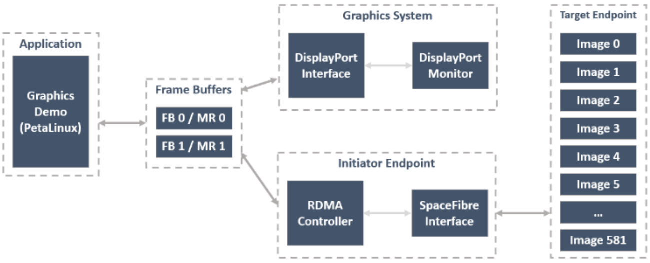

Fig. 3. SpaceFibre Endpoint Image Transfer Demonstration Block Diagram

As shown in Fig. 3, the frame buffers are shared between the Application, Graphics System and Initiator Endpoint. This allows shared memory regions used by the Initiator Endpoint to store images read from the Target Endpoint; used by the DisplayPort Interface to output images to the DisplayPort monitor; and used by the Application to draw statistics.

During initialisation of the demonstration, the set of 1440p images is read from the ZCU102's file system and written sequentially to the 8 GB of DDR3 in the target endpoint using single RDMA write operations.

After the target's images have been initialised, the demonstration application reads the images sequentially directly into alternating DisplayPort frame buffers using single RDMA read operations. After each operation has completed, the application measures and draws statistics at the top of the frame buffer, then issues a synchronisation call to the graphics framework to swap frame buffers after the next refresh.

A photograph of the demonstration's statistics while running is provided in Fig. 4.

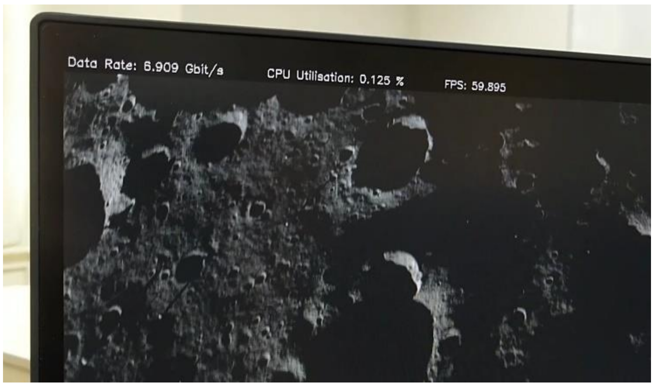

Fig. 4. SpaceFibre Endpoint Image Transfer Demonstration Statistics

In Fig. 4, the photograph shows the statistics being drawn into the frame buffers by the application during the demonstration.

Measuring the processor overhead while running the demonstration application shows a CPU utilisation of less than 0.5% to read the 1440p resolution images from the target over SpaceFibre and maintain output on the DisplayPort interface at 60 frames per second. The data rate is approximately 6.9 Gbit/s, which is the maximum supported by the DisplayPort interface, minus the 32 rows of pixels used for the statistics bar which are not transferred.

V. Conclusions

SpaceFibre provides very high data-rate links using multiple lanes. To maximise link utilisation and minimise processor overhead in embedded systems, it is important to offload as much work as possible from the processor. In order to achieve these objectives, an implementation of RDMA over SpaceFibre was developed, allowing a low-cost path between user-space applications and physical SpaceFibre networks and zero-copy transferring of data.

After initial prototyping of RDMA over SpaceFibre during the Hi-SIDE project that ended in 2022, the SpaceFibre Endpoint was further developed and used in a performance test system and image transfer demonstration. In both cases, the hardware setup included a full SpaceFibre Endpoint implemented in a Xilinx ZCU102 Zynq UltraScale+ board, connected to a target-only SpaceFibre Endpoint implemented in a STAR-Ultra PCIe board, with both boards connected using a 4xSFP-to-QSFP adaptor cable.

Performance results were gathered for various test cases, showing that it is possible to achieve high link utilisation with low CPU utilisation. For example, using a 31.25 Gbit/s link signalling rate, which has a maximum unidirectional data rate of 23.8 Gbit/s, performance results demonstrated over 95% link utilisation with 1% CPU utilisation or less.

In addition to the performance testing, an image transfer demonstration was developed to show RDMA over SpaceFibre being used in a real application. The DisplayPort interface on the ZCU102 was used for this purpose. This supported a maximum of 1440p resolution at 60 Hz, or 4K resolution at 30 Hz, resulting in approximately 7 Gbit/s or 7.9 Gbit/s, respectively. The image transfer demonstration described in this paper used 1440p images, with 582 images stored on the target's DDR, transferred over SpaceFibre, and output on the DisplayPort interface at 60 Hz while using less than 0.5% CPU utilisation.

Acknowledgment

The Hi-SIDE project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 776151.

References

[1] ECSS Standard ECSS-E-ST-50-11C, "SpaceFibre – Very High-Speed Serial Link", Issue 1, European Cooperation for Space Data Standardization, May 2019, available from http://www.ecss.nl

[2] Hi-SIDE Consortium, "Hi-SIDE | High-Speed Integrated Satellite Data Systems", https://www.hi-side.space/

[3] S. Parkes, et al, "SpaceFibre Payload Data-Handling Network", International SpaceWire and SpaceFibre Conference, Pisa, Italy, October 2022.

[4] D. Gibson, et al, "Hi-SIDE: Monitoring, Control and Test Software in a SpaceFibre Network", International SpaceWire and SpaceFibre Conference, Pisa, Italy, October 2022.

[5] STAR-Dundee, "STAR-Ultra PCIe", https://www.star-dundee.com/products/star-ultra-pcie/

[6] Xilinx, "Zynq UltraScale+ MPSoC", https://www.xilinx.com/products/silicon-devices/soc/zynq-ultrascale-mpsoc.html

[7] D. Dunning, et al, "The Virtual Interface Architecture", IEEE Micro, vol. 18, no. 2, pp. 66-76, March-April 1998.

[8] InfiniBand Trade Association, "InfiniBand Architecture Specification", Release 1.6, July 2022, available from https://www.infinibandta.org/

[9] ECSS Standard ECSS-E-ST-50-52C, "SpaceWire – Remote Memory Access Protocol", Issue 1, European Cooperation for Space Data Standardization, February 2010, available from http://www.ecss.nl

[10] Xilinx, "Zynq UltraScale+ MPSoC ZCU102 Evaluation Kit", https://www.xilinx.com/products/boards-and-kits/ek-u1-zcu102-g.html

[11] A. Gonzalez Villafranca, A. Ferrer Florit, M. Farras Casas, S. Parkes, "SpaceFibre IP Cores for the Next Generation of Radiation-Tolerant FPGAs", International SpaceWire and SpaceFibre Conference, Pisa, Italy, October 2022.