REMOTE DIRECT MEMORY ACCESS OVER SPACEFIBRE

Dave Gibson, Andrew MacLennan, Stuart Mills and Steve Parkes

Presenter: Dave Gibson

Conference Themes: Software Architectures and Frameworks, Data Handling and Processing for Data-Hungry Missions and Applications, Networks and Sensors, Embedded Systems

STAR-Dundee Ltd, STAR House, 166 Nethergate, Dundee, DD1 4EE, Scotland, UK.

Email: david.gibson@star-dundee.com

ABSTRACT

SpaceFibre [1] is an on-board network technology for spaceflight applications capable of running at multi Gbit/s and which runs over electrical or fibre-optic cables.

Using multiple lanes, SpaceFibre can scale up the link signalling rate and corresponding data rate. For example, a quad-lane SpaceFibre link running at a lane signalling rate of 6.25 Gbit/s has a link signalling rate of 25 Gbit/s. SpaceFibre currently uses 8b/10b encoding and there is additionally an approximate 4% protocol overhead for unidirectional traffic and 7% protocol overhead for bidirectional traffic. For a 25 Gbit/s link signalling rate, this results in approximately 19.2 Gbit/s unidirectional data rate or 18.6 Gbit/s bidirectional data rate.

The objective of the system described in this paper is to utilise the high data rates provided by SpaceFibre as much as possible in embedded systems with minimal Central Processing Unit (CPU) utilisation. This objective is achieved using a Remote Direct Memory Access (RDMA) based SpaceFibre Endpoint to minimise the amount of work performed by the CPU and provide a low-cost path between user-space software applications and the physical SpaceFibre network.

This paper provides background on the SpaceFibre Endpoint research and development, describes how RDMA is used over SpaceFibre, presents the SpaceFibre Endpoint test system, then provides performance results for the SpaceFibre Endpoint test system.

1 BACKGROUND

STAR-Dundee, in collaboration with partners across Europe, participated in the Hi-SIDE project [2]. Hi-SIDE was a European Union Horizon 2020 project with the aim to develop and demonstrate technologies that enable future high-speed on-board data-handling systems, including a SpaceFibre Payload Data-Handling network [3].

As part of the Hi-SIDE project, research was undertaken to design and prototype a SpaceFibre Endpoint to directly interface SpaceFibre links to embedded processing systems. During this research, an RDMA-based approach was identified as a solution to achieve the high data rates and minimal CPU utilisation required for the SpaceFibre Endpoint.

The initial prototype SpaceFibre Endpoint developed during the Hi-SIDE project was implemented in a STAR-Dundee STAR-Ultra Peripheral Component Interconnect Express (PCIe) board [4]. An accompanying user-space Application Programming Interface (API) and kernel-space PCIe driver was developed for Linux.

Following on from the initial prototype, the SpaceFibre Endpoint was further developed with an Advanced eXtensible Interface 4 (AXI4) version implemented in the Field Programmable Gate Array (FPGA) of a Xilinx ZCU102 Zynq UltraScale+ board [5]. Additionally, a target-only SpaceFibre Endpoint was developed for both ZCU102 and STAR-Ultra PCIe platforms to allow physical memory to be accessed via RDMA without software involvement.

This paper focuses on a SpaceFibre Endpoint test system consisting of a full SpaceFibre Endpoint implemented in a Xilinx ZCU102 board, a target-only SpaceFibre Endpoint implemented in a STAR-Ultra PCIe board, with a user-space API and kernel-space platform driver developed and running on PetaLinux 2022.1 in the ZCU102's ARM Cortex-A53 CPU.

2 RDMA OVER SPACEFIBRE

RDMA is a data transfer model that allows data to be moved directly between memory in remote endpoints using read or write operations executed between an initiator and a target.

The RDMA over SpaceFibre system presented in this paper is based on existing RDMA architectures such as the Virtual Interface Architecture (VIA) [6] 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 [7].

2.1 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.
- RDMA Write: writes data directly from user memory in the initiator endpoint to virtual or physical memory in the target endpoint.

In addition to RDMA operations, the SpaceFibre Endpoint also supports traditional send and receive operations using the same zero-copy approach.

Operations are requested by a user-space application and communicated to the SpaceFibre Endpoint using the user-space API. The SpaceFibre Endpoint then translates the requests from the API into RDMA request packets which are sent by the initiator endpoint to the target endpoint over the SpaceFibre network. The target endpoint receives the RDMA request packets, executes the request, and returns any RDMA response packets to the initiator endpoint. Finally, the initiator endpoint notifies the user-space application using a completion containing the status of the operation.

2.2 Layers

On the initiator side, the following layers are involved:

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

A target 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 operation 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:
  - Application (user-space).
  - STAR-RDMA API (user-space).
  - STAR-RDMA driver (kernel-space).
  - SpaceFibre Endpoint (FPGA).
  - SpaceFibre Interface (FPGA).
- Target-only endpoint:
  - SpaceFibre Endpoint (FPGA).
  - 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.

3 TEST SYSTEM

A SpaceFibre Endpoint test system was developed consisting of a full SpaceFibre Endpoint implemented in a Xilinx ZCU102 board connected to a target-only SpaceFibre Endpoint implemented in a STAR-Ultra PCIe board, providing 8 Gigabytes (GB) of Double Data-Rate 3 (DDR3) memory that can be accessed by the initiator endpoint via RDMA reads and writes.

The SpaceFibre Endpoint utilises the STAR-Dundee SpaceFibre Multi-Lane Interface Intellectual Property (IP) Core [8] to provide a 25 Gbit/s SpaceFibre interface at 6.25 Gbit/s lane signalling rate.

The STAR-RDMA user-space API and kernel-space platform and PCIe drivers were developed for the Linux operating system. In this case, the ARM Cortex-A53 in the Xilinx ZCU102 board is running PetaLinux 2022.1.

A photograph of the SpaceFibre Endpoint test system is provided in Figure 3-1.

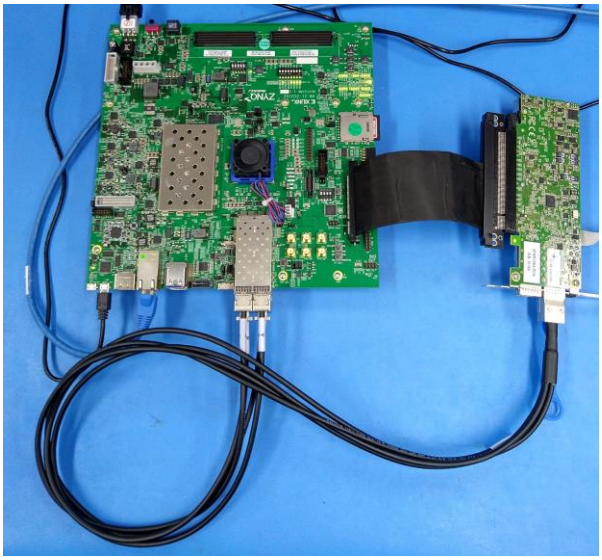

Figure 3-1: SpaceFibre Endpoint Test System

In Figure 3-1, the Xilinx ZCU102 board is on the left, and the STAR-Ultra PCIe board is on the right. For power, the STAR-Ultra PCIe is connected to the ZCU102 via a PCIe extender cable. The SpaceFibre link

is provided by connecting the four Small Form-Factor Pluggable (SFP) interfaces on the ZCU102 to the Quad SFP (QSFP) interface on the STAR-Ultra PCIe using a 4xSFP-to-QSFP adaptor cable. Additionally, there is a serial connection and Ethernet connection to a development Personal Computer (PC) for remote access to the ZCU102.

4 PERFORMANCE RESULTS

To measure performance of RDMA over SpaceFibre, a performance test application was developed in C++ to run custom test cases and measure data rates and CPU utilisation.

In the results presented in this section, the data rate is defined as the bits per second of user data transferred between endpoints. The CPU utilisation is defined as a percentage of the CPU utilisation across the four cores of the ARM Cortex-A53, including the operating system and any other background processes.

4.1 Single Operation Performance

The simplest performance test is to measure performance of single RDMA operations of varying sizes executed sequentially over time.

Each single operation test case, varying from 4 Kilobytes (KB) to 512 KB data length, was executed for 100 iterations with 10 seconds per iteration. Each result is listed as the average followed by the worst-case in parentheses.

The single RDMA write results are listed in Table 4-1.

Table 4-1: Single RDMA Write Results

Test Case	Data Rate (Gbit/s)	CPU Utilisation (%)
4 KB	2.12 (2.11)	16.18 (17.19)
8 KB	4.12 (4.10)	15.59 (16.79)
16 KB	6.78 (6.77)	13.50 (14.00)
32 KB	10.16 (10.14)	10.24 (13.50)
64 KB	13.49 (13.46)	6.46 (7.18)
128 KB	16.11 (16.09)	3.74 (4.01)
256 KB	17.62 (17.61)	2.05 (2.57)

The single RDMA read results are listed in Table 4-2.

Table 4-2: Single RDMA Read Results

Test Case	Data Rate (Gbit/s)	CPU Utilisation (%)
4 KB	1.95 (1.93)	14.48 (15.88)
8 KB	3.53 (3.52)	14.17 (15.55)
16 KB	5.96 (5.95)	10.84 (12.74)
32 KB	9.03 (9.02)	8.66 (9.03)
64 KB	12.31 (12.28)	5.57 (5.98)
128 KB	14.95 (14.95)	1.58 (2.40)
256 KB	16.78 (16.77)	0.86 (1.53)

Executing a single operation involves submitting the operation to a work queue, initiating the operation, and waiting for the completion. Performance can be improved by coalescing operations and completions into groups to reduce the amount of work required per group of operations. The following section provides results for grouped operations.

4.2 Grouped Operations Performance

Each grouped operations test case, varying from 4 KB to 512 KB data length and group sizes from 8 to 128, was executed for 100 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 grouped RDMA write results are listed in Table 4-3.

Table 4-3: Grouped RDMA Write Results

Test Case	Data Rate (Gbit/s)	CPU Utilisation (%)
4 KB x 128	15.16 (15.15)	4.21 (4.47)
8 KB x 128	16.90 (16.89)	3.33 (3.58)
16 KB x 128	17.93 (17.91)	1.65 (1.89)
32 KB x 64	18.40 (18.40)	0.75 (1.00)
64 KB x 32	18.65 (18.65)	0.67 (0.99)
128 KB x 16	18.78 (18.77)	0.48 (0.86)
256 KB x 8	18.84 (18.84)	0.34 (0.42)

The grouped RDMA read results are listed in Table 4-4.

Table 4-4: Grouped RDMA Read Results

Test Case	Data Rate (Gbit/s)	CPU Utilisation (%)
4 KB x 128	14.91 (14.90)	4.73 (4.96)
8 KB x 128	16.74 (16.73)	1.70 (1.89)
16 KB x 128	17.84 (17.83)	2.80 (5.02)
32 KB x 64	18.30 (18.29)	0.66 (0.89)
64 KB x 32	18.53 (18.53)	0.73 (0.99)
128 KB x 16	18.65 (18.64)	0.44 (0.63)
256 KB x 8	18.71 (18.71)	0.28 (0.56)

As shown in Table 4-3 and Table 4-4, data rates and CPU utilisation can be improved significantly, compared to single operation test cases, by coalescing operations into groups.

For groups of operations with a length of 32 KB or above, a data rate in excess of 18 Gbit/s can be achieved with 1% CPU utilisation or less. Compared to the maximum unidirectional data rate possible on a 25 Gbit/s link, which is approximately 19.2 Gbit/s, the link utilisation is approximately 95.3-97.4% calculated using the worst-case results from these test cases.

5 CONCLUSIONS

SpaceFibre is an on-board network technology for spaceflight applications that provides multi Gbit/s data rates. To utilise these high data rates efficiently in embedded systems, it is important to minimise the amount of work performed by the CPU and provide a low-cost path between user-space applications and the physical SpaceFibre network.

This paper presents an implementation of RDMA over SpaceFibre in a test system consisting of a full SpaceFibre Endpoint implemented in a Xilinx ZCU102 board connected to a target-only SpaceFibre Endpoint implemented in a STAR-Ultra PCIe board. The associated software stack includes a user-space API and kernel-space platform and PCIe drivers running in PetaLinux 2022.1 on the ZCU102's ARM Cortex-A53 CPU.

A performance test application was developed to gather results for various test cases, demonstrating that using RDMA over SpaceFibre can achieve high data rates with low CPU utilisation in embedded systems.

6 ACKNOWLEDGEMENT

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

7 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, https://www.hi-side.space/

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

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

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

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

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

[8] Villafranca, A. et al, "SpaceFibre IP Cores for the Next Generation of Radiation-Tolerant FPGAs", International SpaceWire and SpaceFibre Conference, Pisa, Italy, October 2022.