



Dolphin Interconnect Solutions

PCI Express Reflective Memory / Multicast

Whitepaper

DISCLAIMER

DOLPHIN INTERCONNECT SOLUTIONS RESERVES THE RIGHT TO MAKE CHANGES WITHOUT FURTHER NOTICE TO ANY OF ITS PRODUCTS AND DOCUMENTATION TO IMPROVE RELIABILITY, FUNCTION, OR DESIGN. DOLPHIN INTERCONNECT SOLUTIONS DOES NOT ASSUME ANY LIABILITY ARISING OUT OF THE APPLICATION OR USE OF ANY PRODUCT OR DOCUMENTS.

Notes

This document is based on information available at the time of publication. While efforts have been made to be accurate, the information contained herein does not purport to cover all details or variations in hardware and software.

Trademarks

SCRAMNet is a registered trademark of Systran Corporation.

GE FANUC is a registered trademark of GE Fanuc Automation Inc.

Windows is a registered trademark of Microsoft Corporation.

Table of Contents

| | |
|--|----|
| DISCLAIMER | 1 |
| Notes | 1 |
| Table of Contents | 2 |
| Introduction | 3 |
| Multicast implemented in hardware..... | 3 |
| Traditional reflective memory | 3 |
| PCI Express reflective memory | 4 |
| Multicast memory and multicast groups | 5 |
| Using PCI Express reflective memory | 5 |
| Transmitting data to reflective memory | 5 |
| Reading Data from reflective memory | 6 |
| Interrupts..... | 6 |
| Significant benefits provided by PCI Express | 6 |
| Performance | 7 |
| Hardware configuration and installation..... | 7 |
| Reflective Memory Comparison..... | 8 |
| Roadmap and future plans | 9 |
| SISCI API..... | 9 |
| SISCI API Code examples | 9 |
| Reference and more information..... | 10 |

Introduction

The Dolphin Express MX and PX product families together with our eXpressWare software suite supports multicast operations as introduced by the PCI Express Base Specification 2.1. Dolphin has integrated support for this functionality into the SISI API (**S**oftware **I**nfrastructure **S**hared-Memory **C**luster **I**nterconnect) specification to make it easily available to application programmers. The combination of Dolphin PCI Express hardware and the eXpressWare SISI API creates a solution for customers seeking multi-cast or reflective memory type functionality.

The first Dolphin Express product line was introduced in 1994 and has been followed by several generations of shared memory solutions. The Dolphin Express MX product is our fourth generation of interconnect products supporting a real hardware-based multicast implementation. PCI Express multicast enables a single transaction to be sent to multiple remote targets or in PCI Express technical terms - multicast capability enables a single TLP to be forwarded to multiple destinations.

Dolphin combines PCI Express multicast with our SISI API. The combination allows customers to easily implement applications that directly access and utilize PCI Express' reflective memory functionality. Now, applications can be built without the need to write device drivers or spend time studying PCI Express chipset specifications.

The advantage of the PCI Express reflective memory approach is lower latency and higher bandwidth. Dolphin benchmarks show end-to-end latencies as low as 0.7 microseconds and over 27 Gigabytes /sec dataflow at the application level. These benchmarks are included in the SISI developer's kit. By using PCI Express based reflective memory functionality, customers can easily solve their real time, distributed computing performance requirements.

Multicast implemented in hardware

Reflective memory systems (in computer literature also referred to as mirror memory systems, replicated shared memory, multicast or replicated memory systems) implement transparent and automatic updates of remote memory areas. Reflective memory is typically mapped into an embedded system application and enables similar applications on other nodes to share updated data without involving any traditional networking protocol and overhead. Data of any size is automatically transmitted to all nodes directly by functionality implemented in hardware.

Typical applications can range from a two-node fail over pair to large distributed shared memory applications like aircraft, ship and submarine simulators, automated testing systems, industrial automation, electronic trading, control, online and high-speed data acquisition and distribution. Because of their inherent replication they are especially good for fault tolerance.

Traditional reflective memory

Other reflective memory type solutions typically implement reflective memory by providing a plug-in adapter card with onboard device memory. Applications can write to this memory and the data is automatically forwarded through to all other nodes connected. Applications reads data from the local adapter card device memory. A ring network topology connects the systems together. A typical 4 / 8 node configuration can be seen in the figure below.

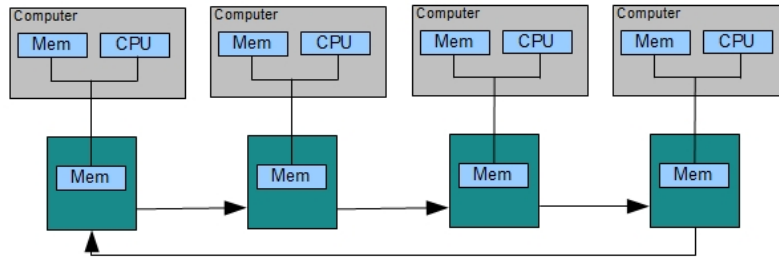


Figure 1 : Alternative types of reflective memory implementation

PCI Express reflective memory

The Dolphin solution is unique as it is able to utilize the computer system’s standard main memory. This, combined with regular PCI Express technology running at wire speeds up to 256 GT/s gives significant performance improvements.

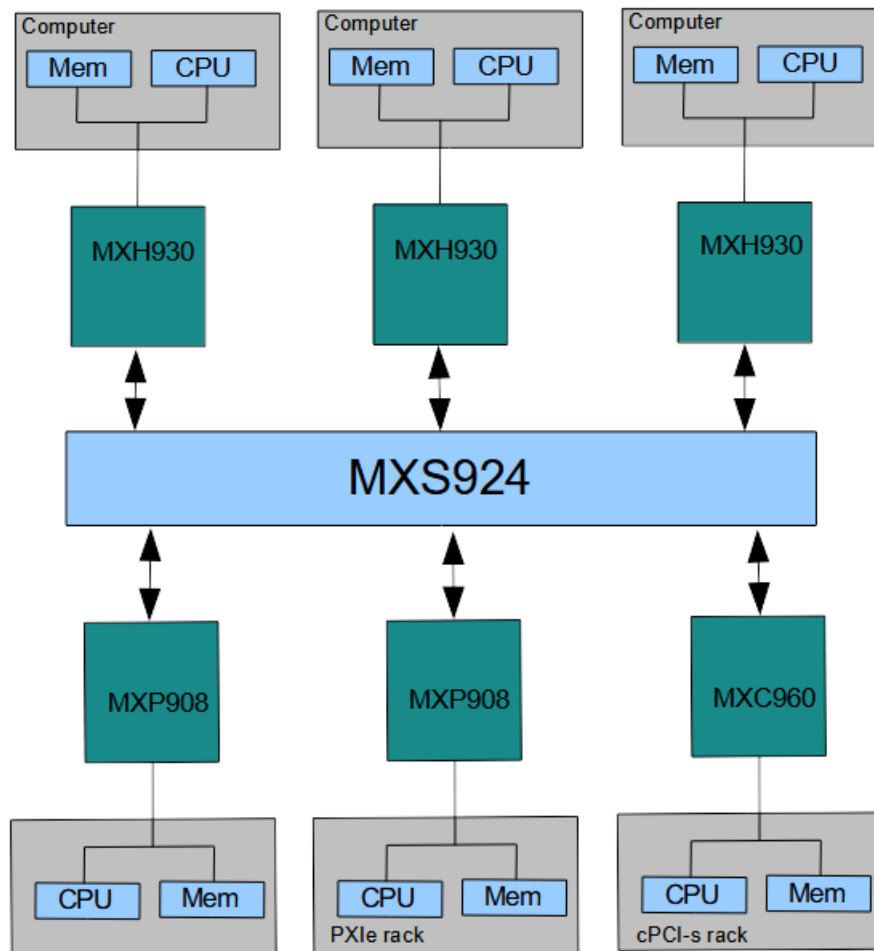


Figure 2 Dolphin Express MX reflective memory setup

The figure above visualizes a typical Dolphin Express setup. Computers can be a mix of supported x86_64, ARM SoCs, PXIe and Compact PCI Serial racks. Each computer has Dolphin PCIe NTB host adapter installed

and they are interconnected through a PCI Express switch fabric. The host adapters do not have any memory used for storing reflective memory data, resulting in significant performance and cost benefits. The PCIe switch (in this example, the MXS924) provides a mechanism for simultaneous multi-cast of data to all connected ports with a measured port to port latency less than 130 nanoseconds.

Multicast memory and multicast groups

The eXpressWare default configuration supports up to 4 independent global multicast groups / memory segments (The maximum number of multicast groups depends on the chipset). This enables SISC I applications to use up to 4 independent reflective memory regions and control which nodes receive the multicast data. This differs from other reflective memory solutions from other vendors which only support a single multicast group.

If a multicast group segment does not exist in a system, the multicast data will be silently dropped without any notification. Multicast data filtering is done by each connected adapter. Nodes can be rebooted and multicast segments can be added or removed at any time without any synchronization with the other nodes. Each multicast segment can be up to 2 Gigabytes with a total reflective memory size of 8 Gigabytes. The eXpressWare default max is 4x 64 Megabytes, please contact Dolphin for instructions to increase the max size.

A PC server with large PCI BARS is required to support the 8 Gigabyte option. If you would like to use reflective memory segments larger than 256 Megabytes, you should ask your system vendor to confirm the system BIOS supports memory mapped I/O above 4GB (large Base Address Register support per the PCIe specification).

Using PCI Express reflective memory

The major difference between traditional reflective memory solutions and PCI Express' approach to reflective memory is that the PCI Express solution utilizes two different addresses, one for reading and a different address for writing data. The SISC I API provides these addresses during initialization. The write address is inside the PCI Adapter address space. Any write to this address space will typically trigger an address translation inside the PCI adapter and cause PCIe transactions to be sent to the MXS924 switch and other nodes. The result of reading this address is undefined.

Transmitting data to reflective memory

Data can be transferred to other nodes using the reflective memory solution in the following ways:

- CPU: Data can be sent to reflective memory using one or more CPU posted write instructions. Using SISC I, applications can use the standard `memcpy()` using the reflective memory as a target or do a regular pointer assignment to transmit data. The fully hardware based memory mapped data transmission does not rely on any operating system service or kernel driver functionality and provides the best possible deterministic data transmission latency and jitter.
- PCIe device: customers can use the SISC I API to configure and enable GPUs, FPGAs etc. (any PCIe master device) to send data directly to reflective memory. (Avoiding the need to first store the data in local memory).
- Onboard DMA: The Dolphin Express MXH and PXH adapter cards include an efficient scatter / gather DMA engine that can be engaged to send small or larger amounts of data to reflective memory.

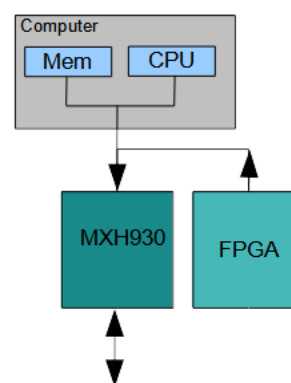
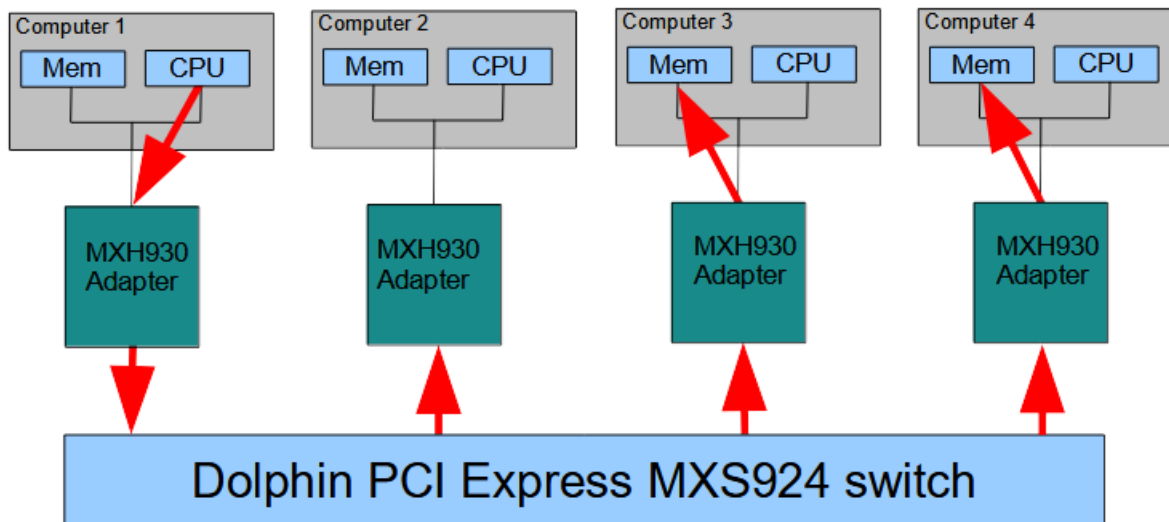


Figure 3: FPGA direct transmission

The figure below shows the flow of data (indicated by the red arrow) – from the CPU of computer 1 - to a local memory address allocated for a specific reflective memory group ID. Data will be transmitted by the PCI Express hardware into the main memory of all other nodes in the network that has allocated a reflective memory segment for the same group ID. All of this is easily managed through the SISC I API. In this example group ID includes computers 3 and 4, does not include computer 2.



Reading Data from reflective memory

To read data received from other nodes, the application needs to use the read address, this points to the allocated segment in local main memory.

If a local reflective memory update is needed, application programmers need to copy the sent data to the local buffer as well. This is a very low cost operation as the data is already in the CPU cache.

Interrupts

The SISI API provides functionality to register and trigger application interrupt's into one or multiple remote nodes. Please consult the SISI Users guide for details on using SISI interrupts.

Significant benefits provided by PCI Express

The PCI Express based reflective memory solutions provides significant improvements over alternative solutions:

- Data in main memory: PCI Express based reflective memory solutions utilize main memory to store data. This has several significant benefits:
 - Reading data in main memory is significantly faster than solutions storing data in specialized PCIe device memory located in the computer IO system.
 - Main memory is cached: This means that the solution will benefit from the standard CPU cache when reading data. Reflective memory updates from remote will automatically invalidate the CPU cache and ensure full data consistency.
 - Specialized device memory is normally very expensive vs main memory modules.
 - You don't need to specify the reflective memory size when buying hardware. The size of PCI Express reflective memory is user configurable – a property set by the application during initialization of the system.
- Data is multicast by a centralized switch.
 - Each PCI Express switch will send data out on all connected ports simultaneously. This means that all nodes will receive data virtually simultaneously when connected to a single switch. When multiple switches are used, each switch hop will add less than 130 nanoseconds delay to the distribution of the data.

- Alternative solutions using a ring topology to distribute data have significant delays between when the first and the last node in the network receives the data. Each node will typically introduce a fixed delay; the total delay in the network varies depending on the number of nodes.
- The minimal delay introduced by PCI Express based reflective memory enables real-time applications to benefit from a significantly reduced total communication time – allowing the application to run at a faster simulation frequency or spend more time on computation.
- Dead nodes or unplugging cables will not stop the entire network; all nodes that remain connected to the network will be able to communicate without interruption.
- Hardware based CRC and retransmission. PCI Express implements a reliable data transmission by calculating a CRC for every data packet. Correctable link errors will automatically cause a hardware retransmit.
- Fair arbitration and sharing of bandwidth. Hard real-time systems should normally be configured to avoid narrow bottlenecks in the network. PCI Express uses a fair, round robin allocation of resources and provides a very deterministic data transmission even under maximum load.
- Protection. The programmer creating the reflective memory application can easily ensure only selected nodes can make updates to the reflective memory or parts of the reflective memory. (E.g. some nodes are only allowed to make updates, other nodes are only allowed to read (portions) of the memory.)

Performance

The performance of the PCI Express reflective memory system mostly depends on the wire speed of the selected PCI Express hardware.

The Dolphin PXH830 PCIe 3.0 x16 adapter used with the MXS824 PCIe 3.0 switch utilizes standard x16 PCI Express link enabling customer applications to take advantage of 128GT/s link bandwidth.

The Dolphin MXH930 PCIe 4.0 x16 adapter used with the MXS924 PCIe 4.0 switch utilizes standard x16 PCI Express link enabling customer applications to take advantage of 256GT/s link bandwidth.

Dolphin reflective benchmarks included in the SISCO developer's kit can be used to measure the reflective memory performance of your system. The actual performance will slightly vary dependent on the computers IO system.

The SISCO reflective memory example 'reflective_bench' can be used to measure the throughput vs message block size. The program is included in the eXpressWare software distribution package.

Hardware configuration and installation

To create a reflective memory system with Dolphin products, each node must have a Dolphin PCI Express NTB adapter card installed in each server/SoC.

The PXH830 adapter cards and the MXS824 switch are compliant to the PCI Express external cable specification 3.0 (PCIe 3.0 cable). The cables are modular based on x4 SFF-8644 connectors. Two cables can be used to create a PCIe x8 link and four cables can be used to create a x16 link.

The MXH930 adapter cards and the MXS924 switch are compliant to the PCI Express external cable specification 4.0 (PCIe 4.0 cable). The cables are modular based on x4 SFF-8644 connectors. Two cables can be used to create a PCIe x8 link and four cables can be used to create a x16 link.

A Dolphin MXS924 or MXS824 switch can be used to connect multiple systems. Up to 6 systems can be connected at x16 speed using a single MXS824 or MXS924. Up to 12 systems can be connected at x8 speed using a single MXS824 or MXS924. Up to 24 systems can be connected at x4 speed using a single MXS824 or MXS924.

For larger reflective memory systems, MXS924 or MXS824 switches are cascaded to create larger topologies. Please refer to the actual software release note for configuration details. The full reflective memory functionality is only available when an MXS924 or MXS824 switch is connected. Two adapter cards can communicate using a direct cable using the standard SISCi unicast functionality (write to only one remote node).

SISCi reflective memory support is targeted at Linux, Windows, VxWorks, QNX or RTX operating systems. The nodes can run any of the above operating systems and inter-communication between Linux, Windows, VxWorks, QNX and RTX systems is fully supported.

PCI Express reflective memory is not limited to server nodes. PCI Express devices are also supported. Data from attached GPUs, FPGAs etc can be distributed to multiple remote nodes simultaneously by simply attaching the PCIe device to a regular PCI Express slot in any of the computers. Additional information can be found in the 'reflective_device.c' example program included in the Dolphin software distribution package.

Reflective Memory Comparison

The various reflective memory systems available have different properties. Generally, PCI Express based reflective memory has significant lower latency, higher throughput but currently supports less nodes and distance. Details for some popular reflective memory solutions can be found in the table below.

| | <i>MXH530*</i> | <i>MXH930</i> | <i>PXH830</i> | <i>PXH820</i> | <i>GEFanuc</i> | <i>SCRAMNet GT</i> |
|--|--|--|-------------------------------------|-------------------------------------|--|--|
| Standard | PCI Express 5.0 | PCI Express 4.0 | PCI Express 3.0 | PCI Express 3.0 | Proprietary | Proprietary |
| Network speed | Max 512 GT/s | Max 256 GT/s | Max 128 GT/s | Max 128 GT/s | 2.12 Gigabit/s | 2.5 Gigabit/s |
| Network topology | Switch – MXS524 | Switch – MXS924 | Switch – MXS824 | Switch – MXS824 | Ring | Ring |
| Max nodes | 30 | 30 | 24 | 24 | 256 | 256 |
| Max distance end-to-end with switch | Copper 2 meters Fiber 100 meters | Copper 2 meters Fiber 100 meters | Copper 3 meters Fiber 100 meters | Copper 3 meters Fiber 100 meters | Up to 10 km | Up to 30 km |
| Cables | SF-8644 copper or Fiber | SF-8644 copper or Fiber | SF-8644 copper or Fiber | SF-8644 copper or Fiber | Fiber | SFP copper or fiber |
| Data Deliver Jitter | 100 ns per switch | 120 ns per switch | 170 ns per switch | 170 ns per switch | 1 us pr node | Less than 1 us pr node |
| 6 nodes | 0.7 us | 0.8 us | 0.7 us | 0.7 us | 8 us | < 8 us |
| 12 nodes | 0.8 us | 0.9 us | 0.7 us | 0.7 us | 14 us | < 14 us |
| 24 nodes | 0.9 us | 0.9 us | 0.7 us | 0.7 us | 20 us | < 20 us |
| Transfer methods | PIO, DMA, PCIe master | PIO, DMA, PCIe master | PIO, DMA, PCIe master | PIO, DMA, PCIe master | PIO, DMA | PIO |
| Write performance PIO | Up to 40 Gigabytes/s | Up to 19 Gigabytes/s | Up to 8 Gigabytes/s | Up to 5 Gigabytes/s | 26 Megabytes/s | 210 Megabytes/s |
| Write performance DMA | Up to 54 Gigabytes/s | Up to 27 Gigabytes/s | Up to 12 Gigabytes/s | Up to 6 Gigabytes/s | 170 Megabytes/s | NA |
| Read performance PIO | 20 Gigabytes/s *2 | 20 Gigabytes/s *2 | 20 Gigabytes/s *2 | 20 Gigabytes/s *2 | 6 Megabytes/s | NA |
| Number of multicast groups | 4 default, max 16 | 4 default, max 16 | 4 default, max 64 | 4 default, max 64 | 1 | 1 |
| Default Memory Configuration | 4 x 2 Megabytes | 4 x 2 Megabytes | 4 x 2 Megabytes | 4 x 2 Megabytes | 256 Megabytes | 128 Megabytes |
| Max Memory configuration | 16 x 128 Gigabytes | 16 x 128 Gigabytes | 64 x 128 Gigabytes | 64 x 128 Gigabytes | 256 Megabytes | 128 Megabytes |
| Type of Memory | System main memory | System main memory | System main memory | System main memory | Device memory | Device memory |
| Fixed memory settings | No, software / DIP-switch configurable | No, software / DIP-switch configurable | No, software configurable | No, software configurable | Yes, card is ordered with a specific memory size | Yes, card is ordered with a specific memory size |
| Memory is cacheable | Yes | Yes | Yes | Yes | No | No |
| Remote interrupts | Yes | Yes | Yes | Yes | Yes | Yes |

1) PCIe 5.0 numbers estimated 2) Depends on system memory bandwidth

The data in the table is found by googling for “reflective memory” and SCRAMNet. Please let us know if the data is incorrect. Actual throughput depends on the local system memory to memory bandwidth.

Roadmap and future plans

Dolphin's reflective memory solution utilizes the standard multicast functionality as defined by the PCI Express Base Specification 2.1 and newer. Upcoming PCI Express 5.0 and future PCI Express 6.0 chipsets will further increase the performance and scalability for applications utilizing PCI Express multicast. Please contact Dolphin for additional information.

Dolphin is committed to maintain a stable SISCi API to enable customers an easily upgrade to new future PCI Express based multicast solutions.

SISCi API

The SISCi API (**S**oftware **I**nfrastructure **S**hared-Memory **C**luster **I**nterconnect) consists of driver and API software, tools, documentation and source needed to develop your own embedded application utilizing the low latency and high performance of a PCI Express Cluster. The SISCi API provides a C system call interface to ease customer integration of PCI Express over cable solutions.

SISCi enables customer applications to easily and safely bypass the limitations of traditional network solutions, avoiding time consuming operating system calls, and network protocol software overhead. SISCi resources (memory maps, DMA engines, Interrupts etc) are identified by assigned IDs and managed by a resource manager enabling portability and independent applications to run concurrently on the same system.

The SISCi API has been defined in the European Esprit project 23174 as a de facto industry standard Application Programming Interface (API) for shared memory based clustering.

In addition to the reflective memory/multicast functionality, the SISCi API provides functionality to access remote memory for unicast (single remote read or write), Direct Remote DMA (RDMA) using the onboard DMA engine. The API also includes support for sending and receiving remote interrupts and error checking. SISCi also support PCIe peer to peer communication over the PCIe cable.

SISCi API Code examples

The SISCi Developers kit contains several basic code examples to demonstrate the use of SISCi and the reflective memory functionality. A good starting point for reflective memory is "[reflective.c](#)" (click to open the source).

Please consult the SISCi API reference manual for more details.

Reference and more information

Please visit www.dolphinics.com/products/MXH532.html for additional information on products and solutions.

Additional information including the SISC! Users guide and the online SISC! API reference manual can be found at www.dolphinics.com/products/embedded-sisci-developers-kit.html

Additional white papers on the Dolphin Express technology are currently available from www.dolphinics.com/support/whitepapers.html :

Please contact sales@dolphinics.com if you have any questions.