Dolphin Interconnect Solutions

Dolphin Express -
Remote Peer to Peer made easy

Whitepaper

Dolphin Engineering

10/8/2015

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.

Table of Contents

Introduction

PCIe peer-to-peer communication (**P2P**) is a part of the PCI Express specification and enables regular PCI Express devices to establish direct data transfers without the need to use main memory as a temporary storage or use of the CPU for moving data. PCI Express peer to peer communications significantly reduce the communication latency but has until now been limited to single systems.

The Dolphin Express product family supports P2P communication and enables local PCI Express devices and PCI Express devices located on remote systems to establish P2P communication as if all devices were local. A single application can directly control all PCIe devices or parallel applications running on multiple servers can implement a protocol to share the devices.

The Intel Phi, GPUs[1], custom FPGAs, specialized data grabbers, video IO Devices etc are devices that typically will benefit from exploiting remote P2P communication to reduce latency and communication overhead.

Dolphin has integrated support for this functionality into the SISCI API specification to simplify the setup and management of peer to peer transfers. The SISCI software enables applications to use CPU / Programmed IO (**PIO**) or DMA operations to move data directly to or from local or remote PCI Express devices. It is also possible to combine the P2P communication with Dolphins reflective memory functionality causing data to be multicasted to multiple devices transparently.

The SISCI API was first defined in 1998 and enables customers to easily implement applications to directly access and utilize PCI Express functionality without the need to write device drivers or spend time on studying PCI Express chipset specifications.

Dolphin benchmarks included in the SISCI developers kit show end to end latencies as low as 0.74us and over 3500 MegaBytes/sec dataflow at the application level.

Hardware configuration

The typical configuration is a modern regular PC with several PCI Express slots. The IO system needs to support standard PCI Express peer to peer communication. A Dolphin PCIe card is installed in a free PCI Express slot and the device that is to be connected over PCIe, an FPGA in the example below, is installed in another PCI Express slot in the same system. Multiple devices can be installed in each host.

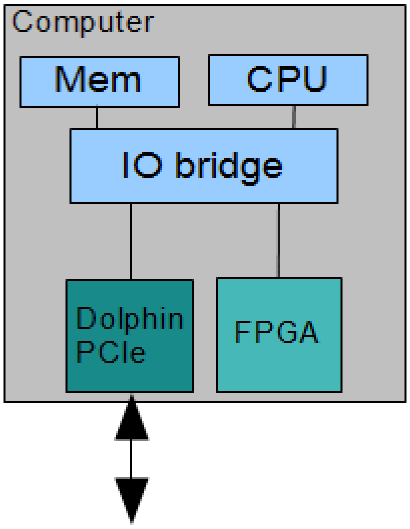


Figure 1 Single node configuration

The FPGA board operates in traditional transparent mode. Depending on the nature of the FPGA and its functionality, the local device driver for the FPGA board needs to be aware of the remote connectivity and sharing. It is up to the designer of this system to solve any sharing issues that may arise between the local device driver and applications accessing the device from a remote system; the SISCI software just enables the sharing functionality.

Connecting computers using PCIe over cable

Two of these systems can be connected directly using a standard PCI Express cable between the Dolphin adapter cards. Several systems can be connected by using a Dolphin Express IXS600 PCI Express switch.

[1] PCI Express does also support GPU Direct™ functionality.

Software configuration

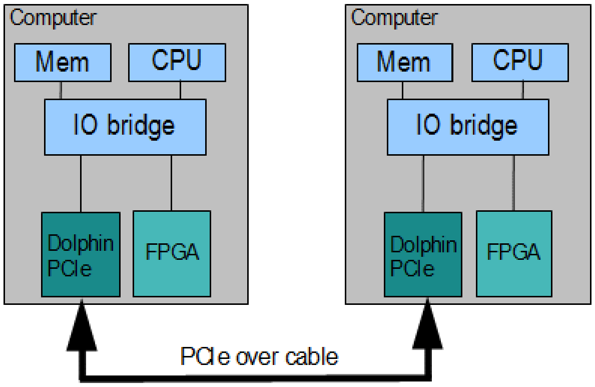


Figure 2 Two nodes interconnected with PCIe

All nodes install the standard Dolphin DIS driver software package. This includes the SuperSockets, The IPoPCIe software and the SISCI API. Only the SISCI API will be used to set up the P2P transfers and the customer needs to develop a SISCI application implementing the desired PCIe peer to peer communication control. The SISCI API provides the mechanisms to ease this implementation.

Basic SISCI functionality is to allocate parts of the system main memory and share it with other cluster nodes. Segments and nodes are identified by a cluster wide unique node IDs and a system wide unique segment IDs. Applications use node IDs and segment IDs to realize connections. The SISCI and IRM drivers (low level drivers, part of the Dolphin driver package) are responsible for safely managing the resources and low level tasks required to establish the connections. NTB mapping tables (LUTs) are set up to perform the local to remote address space translation after appropriate physical addresses are exchanged by the drivers.

Each hardware resource made available over PCIe can be mapped into the controlling applications address space with the appropriate SISCI API functions. Several applications, possibly running on multiple nodes can share these devices, but it it's the responsibility of the application programmer to implement and handle the actual sharing. SISCI provides a rich toolbox for creating clustered applications.

How to make a PCIe address space available for remote access

To register a PCIe device memory as a SISCI segment, application programmers uses the SCIAttachPhysicalMemory() SISCI function to specify the physical address and number of bytes within the PCIe device that should be made available as a SISCI segment. The application also needs to call SCIPrepareSegment() and SCISetSEgmentAvailable(). After these calls have completed, a remote host can connect and map to the physical memory.

How to set up a local PCIe device to access a remote segment or device

To enable a local PCIe device to access a remote SISCI segment (memory or a remote device) you need to identify the corresponding IO address in the local address space. This address can be retrieved using the SISCI SCIQuery() function, flag SCI_Q_REMOTE_SEGMENT_IOADDR by the SISCI application after the remote segment has been connected and mapped. The address returned by the query function can be used directly by the PCIe master to access the remote segment. The address will be inside the BAR address of the Dolphin PCIe card and directly map to the remote address. The customer must make the address available to the PCIe device master. The address is available after the application has completed the SCIConnectSegment() and SCIMapRemoteSegment() functions.

It is also required to register the PCIe device with the Dolphin PCIe card as an approved PCIe master by using the SCIRegisterPCIeRequester() SISCI function. This registration will ensure the master access is passing through the required NTB function.

SISCI Source code

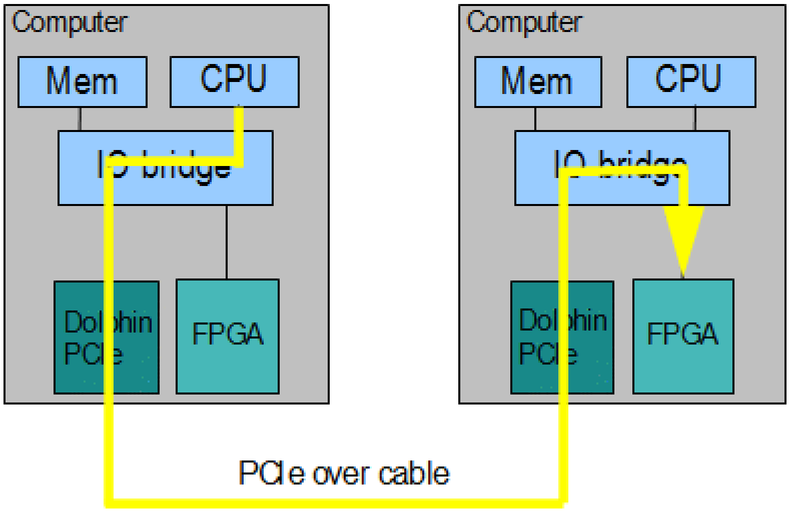
Please review the rpcia.c SISCI test program source code for more details on how to set up a P2P transfer. The program supports both registering a physical device as a segment and to access this from a remote system. The source and binaries are included in the software installers.

Data transfers

The configuration is very flexible and supports concurrent transfers between any of the installed devices and CPU and Memory once the proper connections are established as described above.

CPU or DMA engine used for direct remote access

The figure below visualizes the CPU doing a direct remote access using basic CPU load or store operations.



Larger transfers can be accomplished by using the system bcopy() or moved from local memory to the remote FPGA by engaging the Dolphin PCIe card onboard DMA engine through the appropriate SISCI API function.

FPGA direct access to remote memory

An FPGA device that can act as a PCIe master can directly place data into remote memory by using the address provide by the SCIQuery() function as described above. (Note that slave devices may need special design consideration to achieve the very high source / sink transfer bandwidths that may be desired. However, this is no different than would be required in a single root P2P implementation.)

Multicast

The SISCI software and Dolphin Express IX and PX cluster using the IXS600 switch installed also supports PCIe multicast functionality – often referenced by Dolphin as “reflective memory”. It is possible to combine PCIe multicast and PCIe peer to peer transfers to enable e.g. an FPGA to send data to multiple targets using a single posted write transaction. Please find more details on the reflective memory functionality in the Dolphin reflective memory white paper available from www.dolphinics.com/solutions/whitepapers.html

Interrupt forwarding

Device interrupts, for a device that is accessed from remote, will by default trigger a local interrupt in the system that is hosting the card. The local application that is controlling the device can use regular SISCI API SCICreateInterrupt() and SCITriggerInterrupt() to send interrupts to remote nodes.

Optimized remote interrupt forwarding

Dolphin is planning to offer functionality to enable automatic forwarding of MSI interrupts to remote doorbell registers triggering the SISCI interrupt handler. Please contact Dolphin for further.

SISCI API

The SISCI API (Software Infrastructure Shared-Memory Cluster Interconnect) 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) and Direct Remote DMA (RDMA) using the onboard DMA engine. The API also includes support for sending and receiving remote interrupts and error checking.

Availability

The PCIe peer to peer functionality described above is available with the Dolphin Express IX and PX products. The functionality is available through the SISCI API using Linux, Windows, VxWorks or RTX operating systems. The nodes can run any of the above operating systems – inter-communication between systems running different operating systems is fully supported. The number of transparent devices that can be mapped depends on various diver settings and the size of each device. Please contact Dolphin support for more information and tuning recommendations.

Reference and more information

Please visit www.dolphinics.com for additional information.

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

Please contact pci-support@dolphinics.com if you have any questions.