TECHNOLOGY WHITEPAPER

Enhancing PCIe® Communications to Eliminate Bottlenecks

with Dolphin PCIe Fabric Communications Library

CURTISS-WRIGHT

DEFENSE SOLUTIONS

Read About

Fabric technologies: PCIe, InfiniBand® and Ethernet

Common use of fabrics

PCI Express®

SBC configurations with Dolphin eXpressWare™

Introduction

Today's embedded systems are made up of powerful processing subsystems, each of which is uniquely designed to serve a particular function. In high-performance embedded computing systems, each of these subsystems are often fully functional processing nodes and the limiting factor of system performance often lies in the processor-to-processor data paths. Maximizing throughput and minimizing latency through optimization of the node-to-node data paths will often provide the greatest impact on system performance.

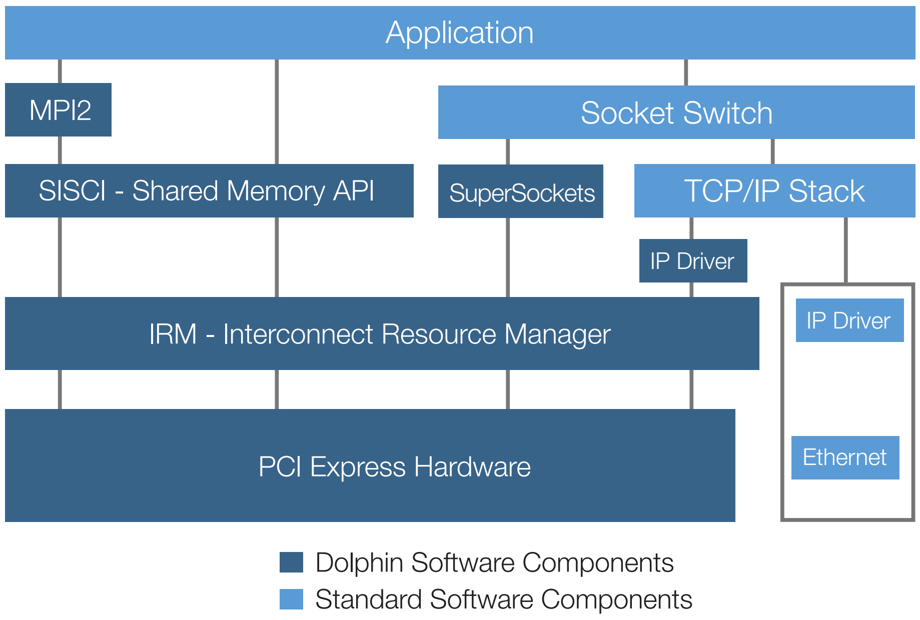
Application
MPI2
Socket Switch
SISCI - Shared Memory API
SuperSockets
TCP/IP Stack
IP Driver
IRM - Interconnect Resource Manager
IP Driver
PCI Express Hardware
Ethernet
Dolphin Software Components
Standard Software Components

Figure 1: Dolphin eXpressWare Software Model

Traditional defense systems built with VME technology have a shared parallel bus, across which all processing nodes communicate. These systems suffer from low overall throughput, not only due to slow data bus transfer speeds, but also because only one node can communicate at a time, further hindering performance. Once a node begins to transfer data, other nodes must wait for their turn to access the shared bus, adding to longer and uncertain latencies.

Modern systems can also use Ethernet to pass data from node to node. With switched Ethernet architectures, nodes can communicate in parallel, virtually eliminating the bottlenecks of a shared bus. However, processor speeds and capabilities have grown much faster than Ethernet speeds, and once again, data paths have become a performance bottleneck.

Almost every contemporary processor today uses the PCI Express® (PCIe) bus as a high-speed interconnect for on-board peripherals. In most processing systems, the PCIe interface offers the fastest data path to and from the processor. Unfortunately, the PCIe bus was never designed to offer straightforward processor-to-processor communications, as its original design based off the PCI parallel bus was for a single "master" host.

Curtiss-Wright Defense Solutions has partnered with Dolphin Interconnect Solutions to bring their eXpressWare™ PCIe fabric software to the embedded VPX world. Uniquely optimized to take advantage of hardware features such as DMA and multi-core processing, eXpressWare can be used to exploit the highest levels of data fabric performance for the rugged defense industry.

In this first of our three part series, we introduce the use of fabrics for high-performance embedded systems, and focus on the hardware and architectural options available to the systems designer. In part two, we compare and contrast several different software interfaces for applications development. We also present performance benchmarks using a variety of Curtiss-Wright 3U VPX modules. Lastly, in part three, we take an in-depth look at device sharing and multicast applications.

Common Uses of Fabrics

Understanding how to use a data fabric may be new to some developers. However, most developers have been using the concept of a data fabric disguised under other names, such as:

+ Memory mapped interconnects
+ Network connected nodes
+ Shared memory subsystems

A fabric can be viewed as any "path" for a processing node to communicate to another processing node. In today's switched fabric systems, the path may be direct from node to node (distributed fabric), or may be indirect through an intermediate or centralized node (centralized fabric).

Different Types of Processor-to-Processor Communications

Messages

In a system with multiple processing nodes, the nodes need to communicate in order to coordinate and synchronize activities. These types of messages are generally short data transfers where latency is critical and overall transfer throughput is not a significant concern. Furthermore, when a message is received, the receiving node needs to be informed of the event through the use of software paradigms such as mailboxes and doorbells.

Data Transfer

For data processing systems, large data transfers are often used between pipeline processing stages or to distribute incoming bulk data for parallel processing and combine the results of parallel processes into a final processing node. The data itself is usually not directly interpreted by the processors at each node; instead the data is simply received and then processed independently with other processes or local hardware accelerators such as GPGPU or FPGA devices.

These bulk data transfers require considerably more time to transfer than short messages and have data buffer sizes often in the tens or hundreds of MB in size. For bulk data transfers, initial latency is masked by the overall data transfer time, and throughput becomes the critical parameter.

Fabric Technologies

InfiniBand

InfiniBand® is a low level fabric similar to SRIO, and has become quite popular in HPEC data centers due to its relatively low latency and high throughput. Similar to SRIO, InfiniBand is now a single vendor technology, with Mellanox® Corporation driving both the InfiniBand standard, and designing InfiniBand interface devices. InfiniBand has garnered limited appeal in deployable defense systems due to limited software support for real-time operating systems, the high cost of silicon devices, and limited support from its single source vendor.

Ethernet

Ethernet has become a universal network interconnect, with wide spread adoption of interface hardware and software support. Almost every processor available today includes support for Ethernet, and all commercial and real-time operating systems support a variety of Ethernet interface protocols, such as TCP/IP and UDP.

Ethernet can support both distributed and centralized interconnect topologies, although centralized architectures with Ethernet switches are almost universally used.

Today's embedded systems often offer Gigabit Ethernet operating at 1 or 10 Gbaud on the control plane and PCIe connectivity on the data plane of 3U VPX boards. As high-performance processors continue to evolve, Gigabit Ethernet operating at 10 and 25 Gbaud will become more common on the control plane, with data plane connectivity offering higher speed 40 and 100 Gbaud interfaces. Similarly, 6U VPX boards often feature Gigabit Ethernet operating at 10 or 40 Gbaud, with 100 Gbaud expected to gain adoption in the coming years. However, many processing systems will still require Ethernet interface devices to be external to the processor, most commonly connected via PCIe.

Typical Ethernet communications is performed with TCP/IP network stacks, which provide feature-rich support for error-free communications and advanced features such as Quality of Service (QoS), multicast and security. These advanced features result in software driver complexities and complex data buffering schemes, greatly increasing latency delays and drastically lowering throughput.

Specialized Ethernet protocols such as RDMA, iWarp and RoCE have been created to bypass some of these limiting software mechanisms and are designed to pass Ethernet data directly into target system application memory buffers. Use of these protocols will not benefit from many of Ethernet's inherent advanced features.

PCI Express

Why PCIe?

PCIe has become the de-facto standard interconnect mechanism for almost all processors today. It often provides the fastest data transfer to and from the processor, and serves as a universal interconnect to other devices for system expansion.

By using PCIe directly and bypassing Ethernet or fabric interface devices, latency is reduced, throughput is increased, and there are other side benefits such as reduced power dissipation, increased MTBF, and lower overall system costs.

PCIe Simplicity and Complexity

Hardware Connection Is Easy

Connecting PCIe devices is extremely easy. The simplest PCIe link is a single lane bi-directional interface with two differential pairs: one transmit and one receive pair. To increase performance, higher lane counts can be made using 2-lane, 4-lane, 8-lane, or 16-lane physical connections.

The PCIe interface supports several data rates. Gen1 interfaces run at 2.5 Gbps, with Gen2 interfaces doubling the data rate to 5.0 Gbps. Gen3 interfaces increase this speed to 8.0 Gbps, and by using a higher efficiency data coding mechanism, the effective data transfer rate becomes double that of Gen2. The development of the PCIe Gen4 standard has recently been completed, and Gen 4 devices are beginning to ship from vendors, again doubling performance with 16 Gbps interconnect speeds. PCIe Gen5 will increase speeds to 32 Gbps and is expected in a few years.

Table 1 summarizes PCIe interface speeds and bus widths using todays (and tomorrow's) PCIe technology.

Table 1 TBD

PCIe Generation	Line Rate (per lane)	Encoding Scheme	Maximum Effective Data Throughput (GBps)			
			1-lane	4-lane	8-lane	16-lane
Gen1	2.5 Gbps	8b/10b	0.25	1	2	4
Gen2	5.0 Gbps	8b/10b	0.50	2	4	8
Gen3	8.0 Gbps	128b/130b	0.98	3.94	7.88	15.75
Gen4	16 Gbps	128b/130b	1.97	7.88	15.75	31.51
Gen5	32 Gbps	128b/130b	3.94	15.75	31.51	63.02

Figure 1: PCIe Performance

Another advantage of PCIe is its ability to automatically negotiate link speeds and lane widths with its partner. When a connection is first made, the partners exchange messages and agree on these interface parameters. During operations, if data communications become unstable, the partners can re-negotiate to support a lower speed or reduced lane width interface, as appropriate.

Root Nodes and Address Space

The PCI interface uses the concepts of a root node and peripheral devices. The root node is almost always at the processor end, and it is the responsibility of the root node to scan the PCI bus and find all connected downstream devices. This process, called enumeration, queries each device found on the PCI bus, determines interface parameters such as memory window requirements of each endpoint device, and assigns resources from within the root node's memory map to satisfy these requirements. After enumeration, the root device can access all connected PCIe peripherals with simple read/write operations within in its local address space.

The PCI interface does not allow multiple root nodes to exist on a single PCI bus because each root node has its own enumeration process and root nodes do not share a common address space. This limitation means two processors, each with independent root nodes, cannot connect directly to one another. Another mechanism is needed to make this connection work.

PCIe Switches

The PCIe interface is a point-to-point interface. In order to connect multiple devices to a single PCIe root node, a PCIe switch can be used to expand the PCIe bus to support multiple downstream devices. For example, a 16-lane PCIe switch may be configured to support one 4-lane upstream port to the processor's root node, and three 4-lane downstream ports to additional peripherals. It may also be configured to support, for example, as many as 12 downstream ports, each of which would be a 1-lane interface, or any mix of supported downstream port sizes. During enumeration, when the root node identifies a switch device, it will scan each of the switch downstream ports to find additional devices.

A switch used in this manner is called transparent, because the upstream root node would be able to enumerate all downstream devices, and all devices would appear in a single transparent address space. The transparent port of the switch does not request PCI bus resources such as memory windows – it is transparent to all PCI bus operations, passing read and write operations through the switch to the correctly addressed nodes.

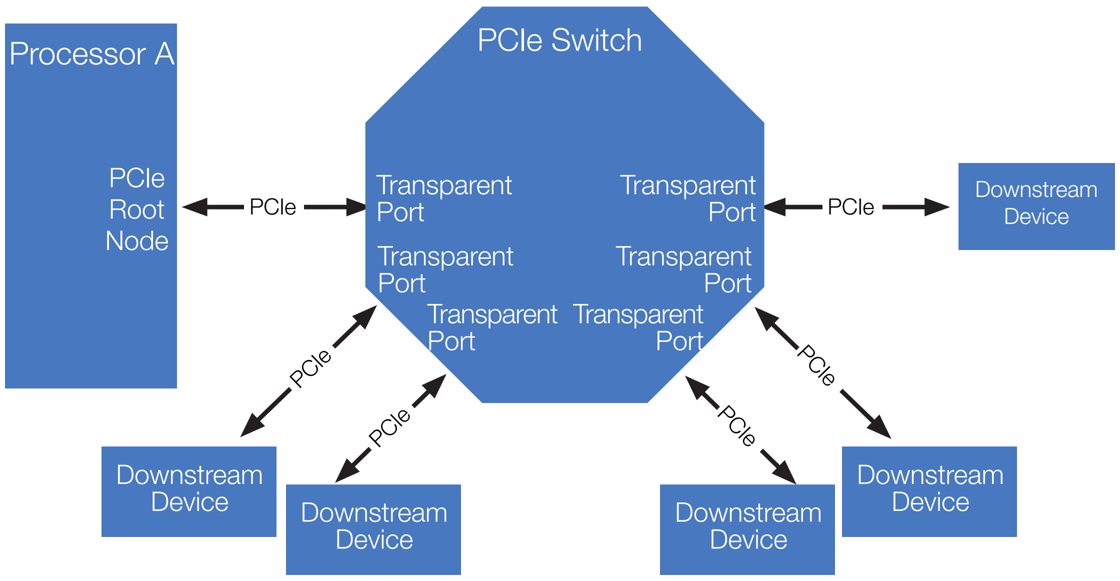

Figure 2: Single Processor System with Transparent PCIe Switch

With this transparent PCIe connection, a root node host still cannot connect to another root node host. The system shown in Figure 3 is invalid because two PCIe root nodes exist in the same PCI address domain.

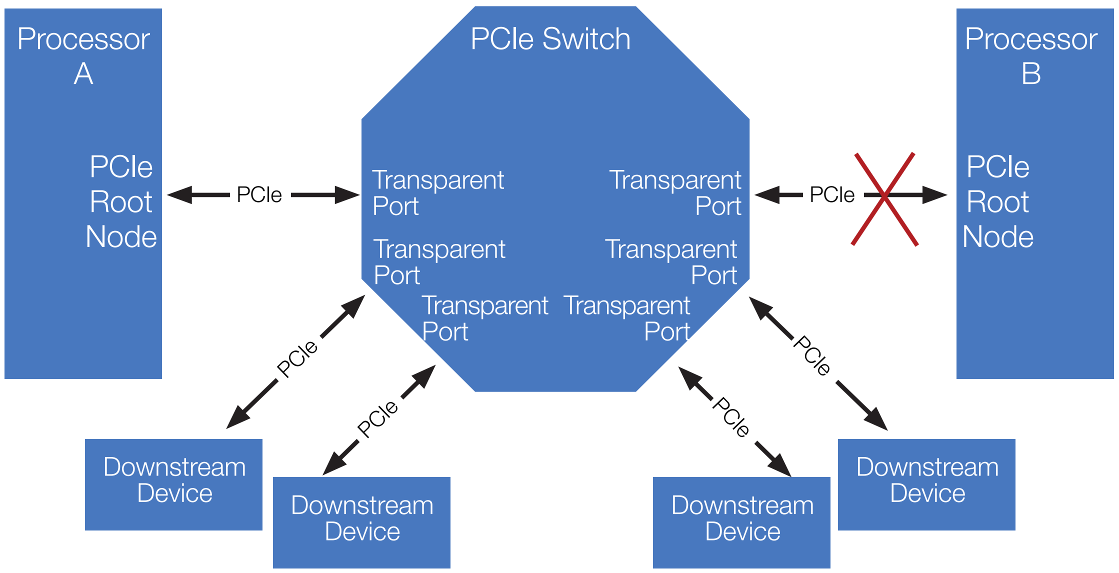

Figure 3: Dual Processor System with Transparent PCIe Switch - Invalid Configuration

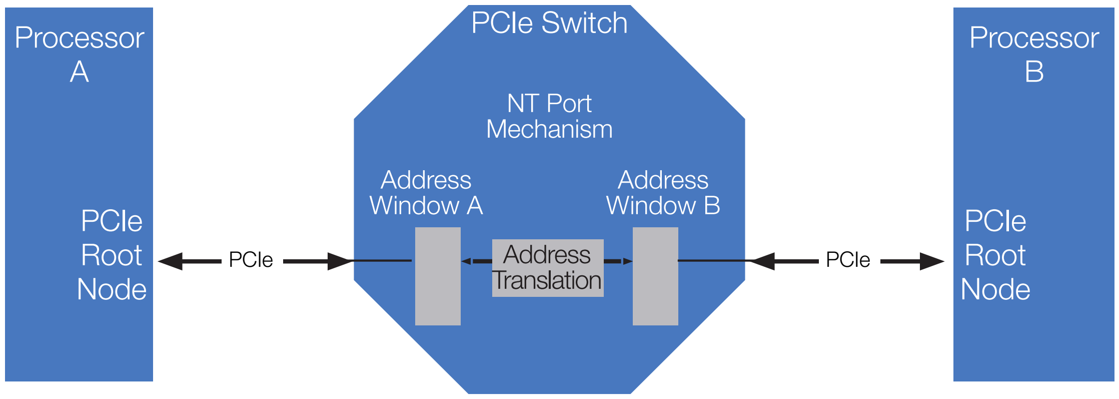

Figure 4: Non-Transparent Port - Address Translation Mechanism

Non-Transparent Ports

To solve the processor-to-processor connection problem, PCIe switches can configure a port as a non-transparent (NT) node. When the root node finds a port configured as NT, it does not look for devices past the NT port, and the enumeration scan does not continue through the NT port. This effectively isolates the PCI bus on each side of an NT port.

The NT port has a mapping mechanism which allows the two processors to communicate (refer to Figure 4). When Processor A enumerates the NT port, the NT port requests a memory region within Processor A's address space. Similarly, when Processor B enumerates the NT port, it also assigns a memory region within Processor B's address space. This common memory region is addressable from either side, but using different addresses from each of the processors. The NT port provides a memory translation mechanism so data written from one side to address A1 appears on the other side as if it was addressed to address B1.

This non-transparent port mechanism allows the two processors, or root nodes, to communicate to one another. Setup and configuration of such a system can sometimes be quite challenging.

Hardware Architectures Supported

Direct Peer-to-Peer Topologies

Board Configurations

The simplest host-to-host architectural model is to directly connect two boards together. An example of this is shown in Figure 5, using two SBC or DSP engines featuring 8-lanes of PCIe connectivity.

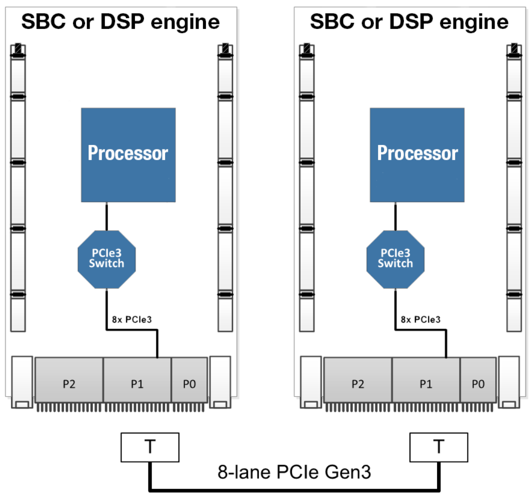

Figure 5: Two-Board Configuration, Default Switch Configurations (Transparent)

In this example, an 8-lane Gen3 PCIe connection is made between two SBCs or DSPs. However, the default configuration of each board's PCIe port is transparent (labeled as T), which connects one PCIe root node directly to another PCIe root node. This configuration violates the PCI bus principal of one root node per PCI bus.

Reconfiguring one of the two PCIe switch ports as non-transparent (NT) will allow two independent PCI buses, one from each board, to exist, with the ability to communicate through the NT port. This is shown in Figure 6 and can be implemented with a single NT between the root nodes, or two NT ports, one at each board.

When the left board enumerates, the process stops at the NT node. Similarly, when the right board enumerates, it also stops at the NT node.

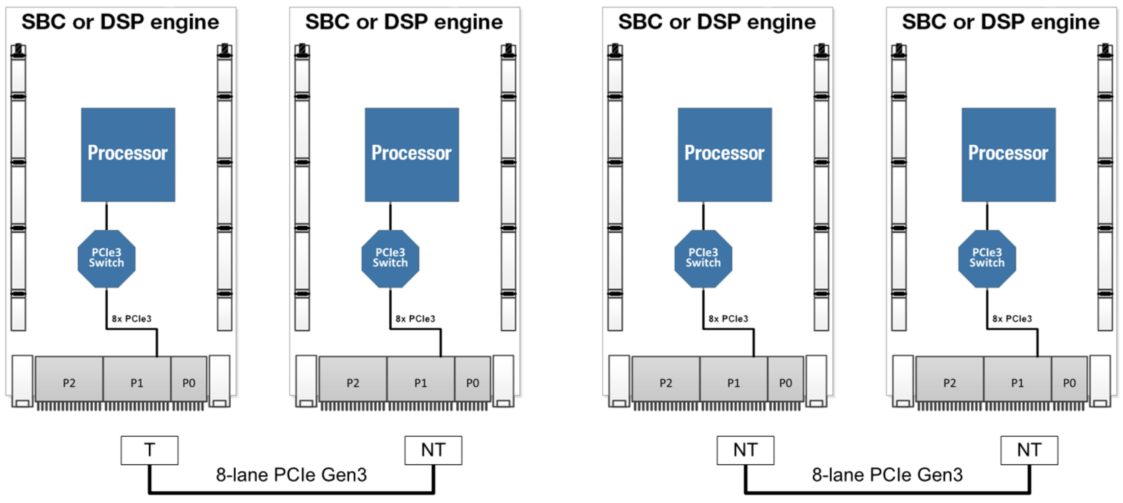

Figure 6: Two-Board Configuration, Alternate Switch Configurations (Non-Transparent)

Board Configurations

In a similar arrangement, a three-board configuration can be setup with 4-lane PCIe connections, with a T to NT connection between boards, as shown in Figure 7. Every process-to-processor PCIe path includes an NT port, and does not violate PCI bus conventions.

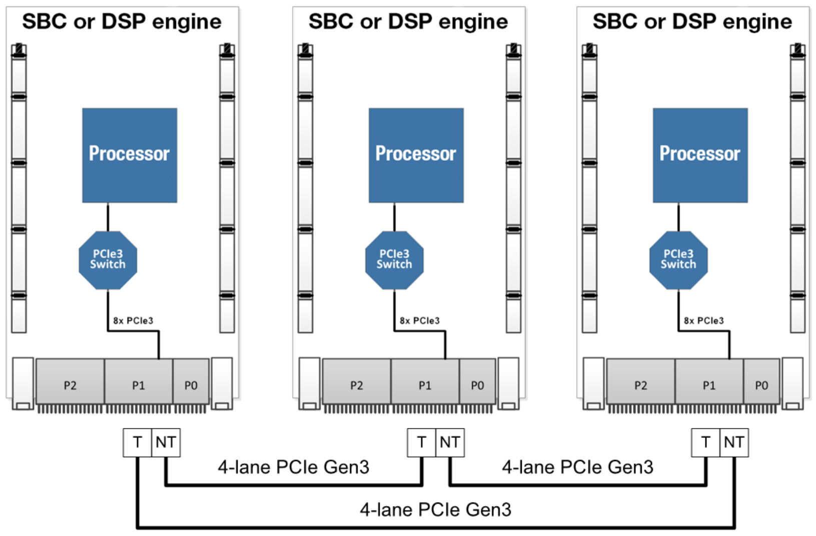

Figure 7: Three-Board Configuration with T to NT PCIe Connections

This configuration provides a direct connection between all boards, allowing any board to communicate to any board directly.

More Board Configurations

Although more than 3 boards can be connected in a similar fashion, this topology becomes increasingly more complex to configure and manage from a software standpoint. Not all nodes will have direct paths to all other nodes. Additionally, many PCIe switches do not permit lane widths below 4-lane, or to support more than one NT port. Another topology is needed to provide full communications within a system with more than 3 boards.

Central Switch Topologies

In a central switch topology, all boards are connected to a central switch, permitting every board to communicate to all other boards. In the example shown in Figure 8, six SBCs are connected to a central switch. The SBC nodes are configured as NT ports and the switch is configured as a T port. Enumeration of each SBC stops at the NT port, isolating each SBC's PCI bus to the local board.

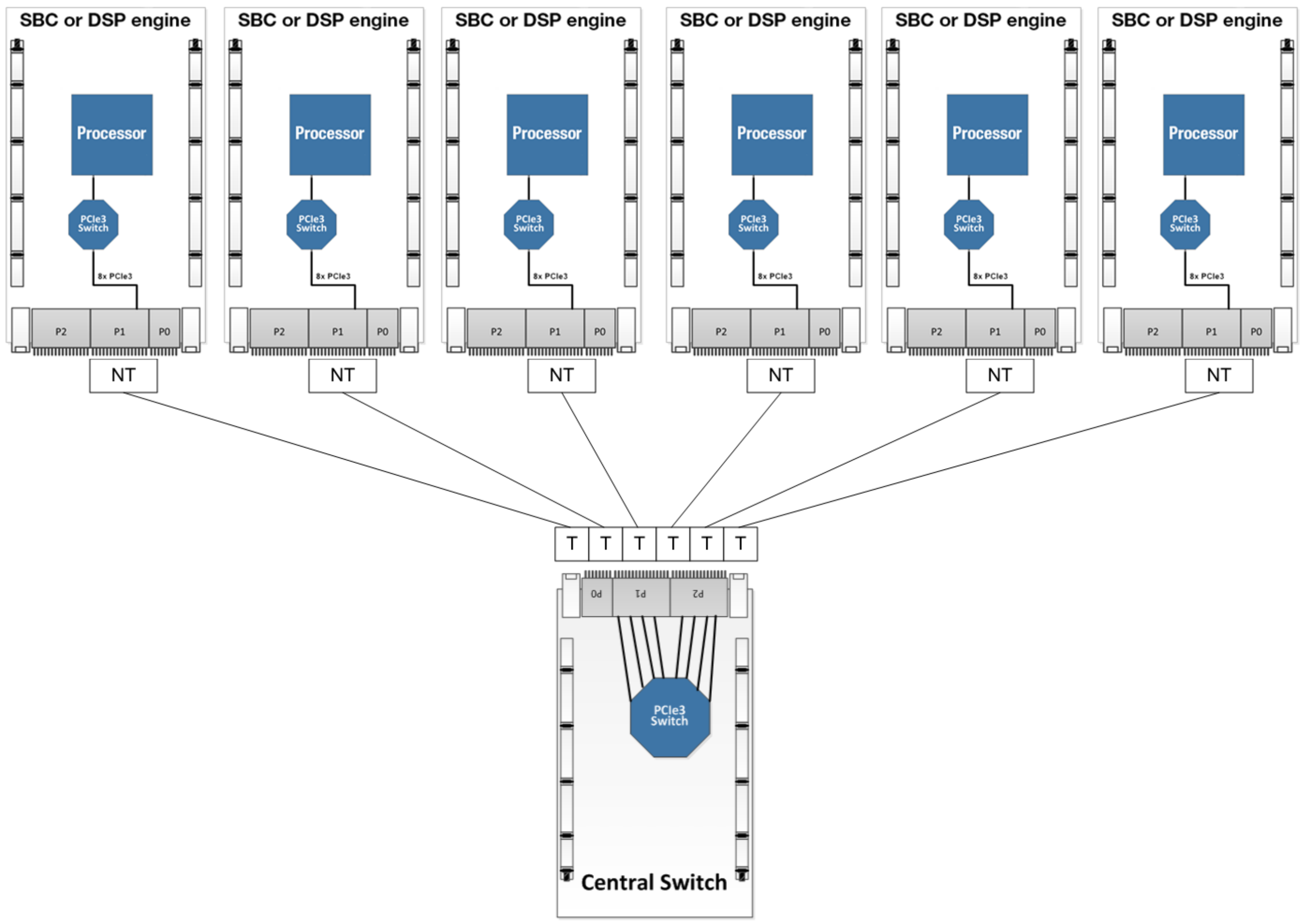

Figure 8: Central Switch Topology

Summary

Rugged embedded applications depend on high performance fabrics to transfer data with low latency and high data transfer rates. In this first part of our Dolphin Fabric Communications white paper series, we introduced the use of fabrics for high-performance embedded systems, and focused on the hardware and architectural options available to the systems designer when using PCIe as a high performance fabric. Curtiss-Wright's partnership with Dolphin and its eXpressWare software enables our customers to have the benefits and flexibility of a more simplified setup for processor-to-processor communications using PCIe connections in a variety of configurations of host-to-host architectures.

In the second part of this white paper series, we will present several different software interfaces provided for applications development, comparing their advantages and tradeoffs. We will also present performance benchmarks using a variety of Curtiss-Wright 3U VPX modules. Finally, in part three we will explore device sharing and multicast applications.

Author(s)

Aaron Frank, BaSC
Senior Product Manager
Curtiss-Wright Defense Solutions

Tammy Carter, MSCS
Senior Product Manager
Curtiss-Wright Defense Solutions

Learn More

Product Sheets

- Dolphin PCIe Fabric Communications Library
- OpenHPEC Accelerator Suite
- VPX3-133 3U VPX SBC
- VPX3-131 3U VPX SBC with NXP Power Architecture P4080 Processor
- VPX3-1220 3U VPX SBC with Intel Xeon 7th Gen Processor
- VPX3-1259 3U VPX SBC with Intel Core i7 Broadwell Processor
- VPX3-1260 3U VPX SBC with Intel Xeon 8th Gen Processor
- CHAMP-XD1/VPX3-482 3U VPX DSP with Intel Xeon D Processor
- XMC-121 3U XMC Mezzanine with Intel Xeon 7th Gen Processor