

Mesh Fabric Switching with Virtex-II Pro FPGAs

Implementing mesh fabric architectures has just gotten easier with the Xilinx Mesh Fabric Reference Design and ATCA Development Platform.

by Mike Nelson

Sr. Manager, Strategic Solutions
Xilinx, Inc.
mike.nelson@xilinx.com

The introduction of the Virtex-II Pro™ Platform FPGA with integrated multi-gigabit transceivers (MGTs) enabled a new era of system design. Specifically, Virtex-II Pro devices now enable designers to implement switched fabric system architectures efficiently, affordably, and entirely in programmable logic.

To illustrate this point and enable its rapid exploitation by our customers, Xilinx developed the Mesh Fabric Reference Design (MFRD), a modular, highly scalable, and configurable resource for building switched fabric system solutions, and the Advanced Telecom Compute Architecture (ATCA) Development Platform. In this article, we'll take a close look at both tools.

Switched Fabric Topologies

The classic switched fabric configuration is a star in which each node communicates with all of the other nodes through a central switch (Figure 1A). The obvious limitation of a star is that it is not fault tolerant. To address this limitation, you need a dual star (Figure 1B).

In a mesh fabric, the switching function is distributed across the system; every node connects directly to each and every other node. This configuration is inherently resilient, as shown in Figure 1C.

To compare the performance of these alternatives, let's consider two atypical 16-slot configurations: a dual star with 10 Gb links, and a mesh with 2.5 Gb links. Because these configurations require approximately the same number of MGT resources for implementation (224 for the star versus 240 for the mesh), they are essentially equal from a power and system cost perspective (i.e., connector and back-plane routing resources).

The maximum theoretical system bandwidth for a dual star is equal to the number of nodes times the link rate times two (as all links are full duplex). In our 16-slot example, this works out to 14 nodes (two slots are required for the switches) x 10 Gb x 2 = 280 Gb.

The maximum theoretical system bandwidth for a mesh is equal to the number of nodes times the number of links per node (nodes minus 1) times the link rate. In our

example, this works out to 16 (all slots are nodes in a mesh) x 15 x 2.5 Gb = 600 Gb.

The mesh configuration is able to achieve more than twice the system performance with essentially equal resources because half of the star is required simply for fault tolerance. Additionally, the star incurs a fractional performance hit because two slots must be dedicated to switching in its chassis, thus limiting the node count.

In fairness, we should note that a dual star can double its theoretical bandwidth to 560 Gb if it uses active-active load balancing, but not with fault tolerance. That would require the addition of a third switch for failover, increase the MGT count to 312, and reduce performance to 520 Gb in a 16-slot chassis, as the node count decreases to 13. Table 1 compares the performance of these configurations, along with additional examples.

Fabric Topology	MGT BW	Link BW	16-Slot Chassis Configuration	
			Aggregate System BW	MGTs Required
4X Star	2.5 Gb	10 Gb	300 Gb	120
4X Dual Star	2.5 Gb	10 Gb	280 Gb	224
Active-Active 4X Dual Star	2.5 Gb	10 Gb	560 Gb	224
A-A 4X Dual Star with HA*	2.5 Gb	10 Gb	520 Gb	312
1X Full Mesh	2.5 Gb	2.5 Gb	600 Gb	240
2X Full Mesh	2.5 Gb	5 Gb	1.2 Tb	480
4X Full Mesh	2.5 Gb	10 Gb	2.4 Tb	960

* Requires three switches

Table 1 – Performance comparison of various star and mesh fabric configurations

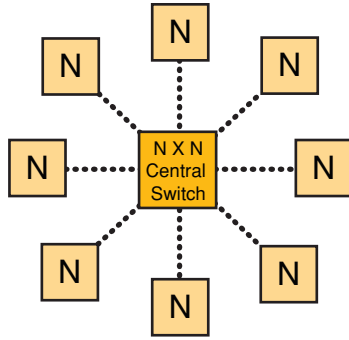


Figure 1A – Star fabric configuration

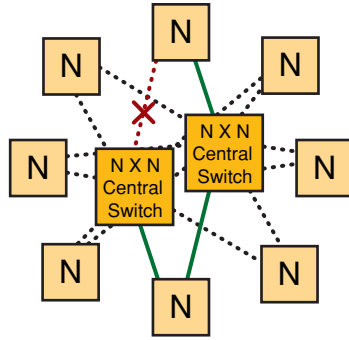


Figure 1B – Dual star fabric configuration

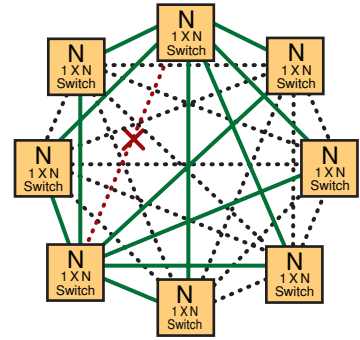


Figure 1C – Mesh fabric resiliency

Figure 1 – Switched fabric topologies

Mesh Fabrics Fit Virtex-II Pro FPGAs

Before the advent of abundant and affordable MGT resources, mesh fabrics were challenging to implement. Now, they're an emerging segment – historically an excellent home for programmable logic.

The *distributed* nature of switching in a mesh fabric enables a mesh to map extremely well to the resources available in Virtex-II Pro Platform FPGAs. These products have everything you need to build exceptional mesh fabric interconnects:

- Four to 24 MGTs per device for implementing serial links
- Block RAM for implementing queues
- Logic for implementing control and traffic management functions
- Embedded PowerPC™ processors that can be used to implement management functions.

Mesh fabrics will also scale well in next-generation Virtex-II Pro X™ Platform FPGAs. The Pro X family introduces 10 Gb MGTs that can quadruple the performance for our 16-slot mesh example to an incredible 2.4 Tb.

The Xilinx Mesh Fabric Reference Design

To enable Virtex-II Pro applications in mesh fabrics, Xilinx developed the Mesh Fabric Reference Design. The MFRD enables an extremely broad range of system configurations.

When designing the MFRD, Xilinx set out to address a number of key objectives:

1. Support system configurations from a few to hundreds of ports
2. Enable flexibility for implementing a chosen configuration and thus the ability to cost-optimize the solution
3. Provide configurable and competent queue management functionality
4. Enable efficient use of fabric bandwidth
5. Support standard Xilinx interfaces on modular boundaries
6. Enable processor-based switch management.

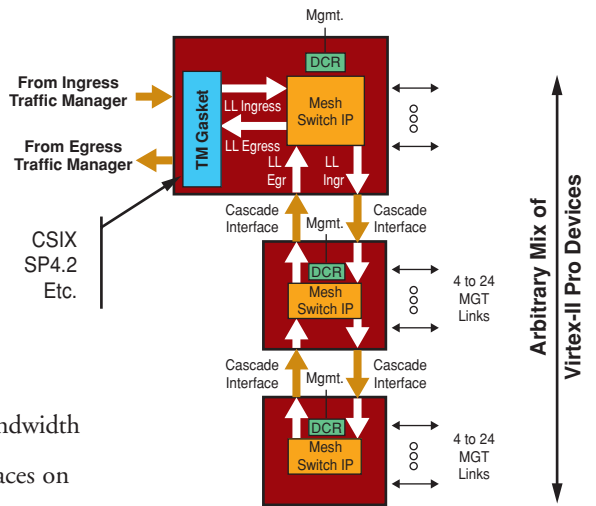


Figure 2 – MFRD architecture

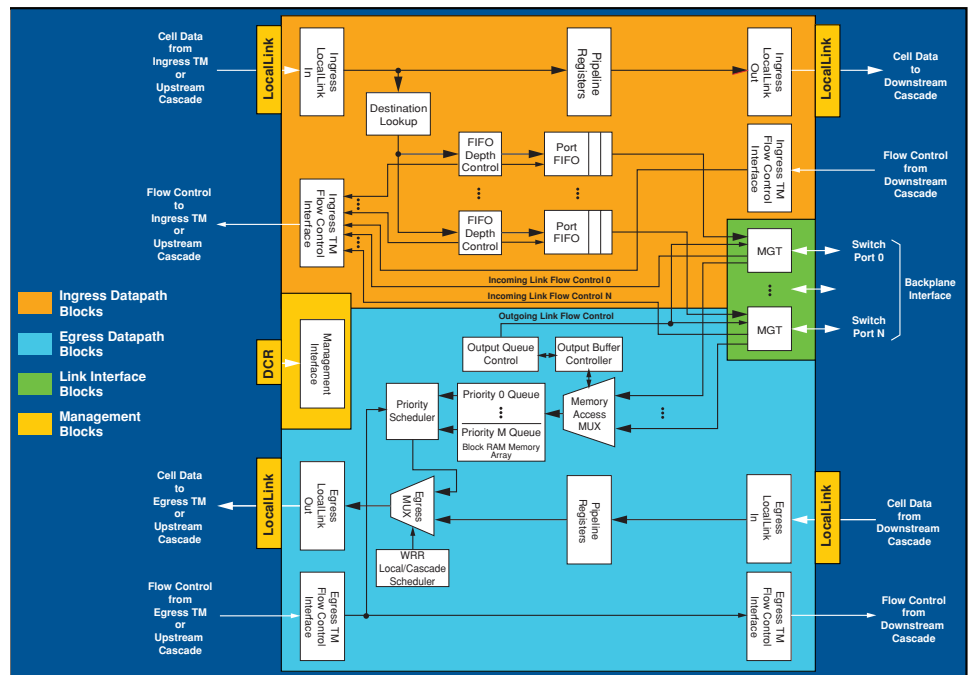


Figure 3 – MFRD block diagram

To achieve these goals, the MFRD implements a mesh switching architecture, as illustrated in Figure 2. The MFRD specifically implements a “mesh switch IP” element illustrated in each device in the figure. We will review the details of this IP, but for now let’s focus on the bigger picture.

The MFRD implements a modular architecture that can be realized in one or more components. This enables configurations from four to 256 ports in any mix of Virtex-II Pro FPGAs and provides designers with exceptional flexibility in configuring their systems. For instance,

you could implement a 16-port switch in a single 2VP50, in a combination of a 2VP20 and 2VP7, or in two 2VP7s. This flexibility is ideal for optimizing the price/performance of the solution to your specific needs.

Other aspects to note in Figure 2 are:

- The use of the standard LocalLink interface for switch ingress and egress
- The use of the device control register (DCR) bus for switch management by the Virtex-II Pro embedded PowerPC RISC processor

- The traffic management gasket: While a key element of any design, it is important to note that this interface will differ for every application and is therefore beyond the scope of the MFRD.

Internally, the MFRD is a cell-based switch architecture supporting 40 to 128 byte payloads. To understand its operation, let’s look at a block diagram and follow the course of traffic from ingress through egress; in this way we can easily understand its features and capabilities. The basic structure of the MFRD is illustrated in Figure 3.

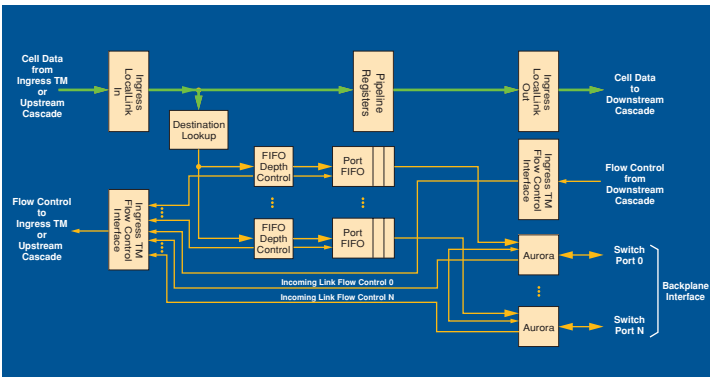


Figure 4A

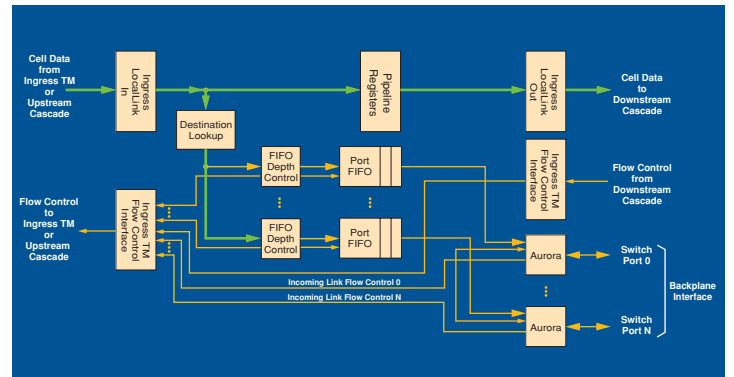


Figure 4B

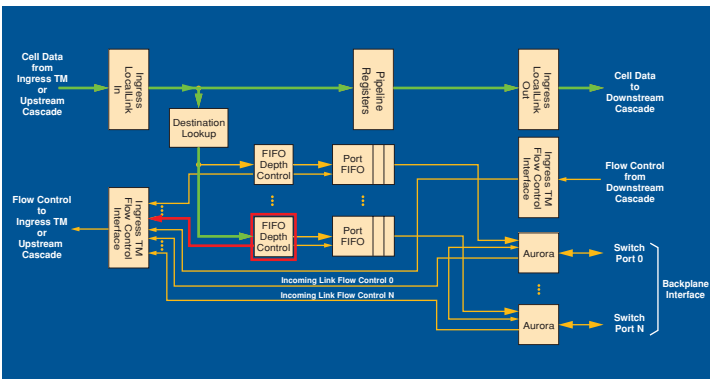


Figure 4C

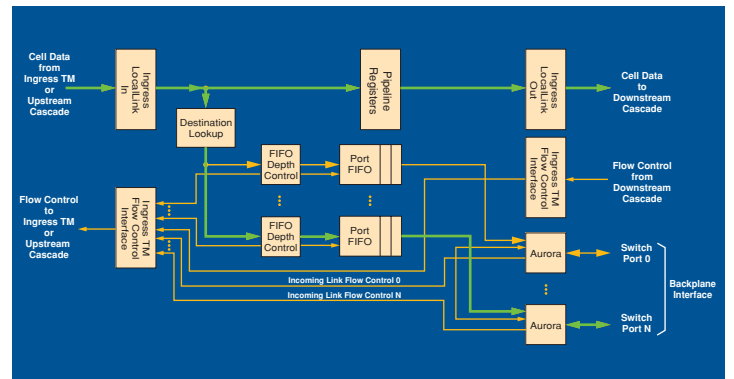


Figure 4D

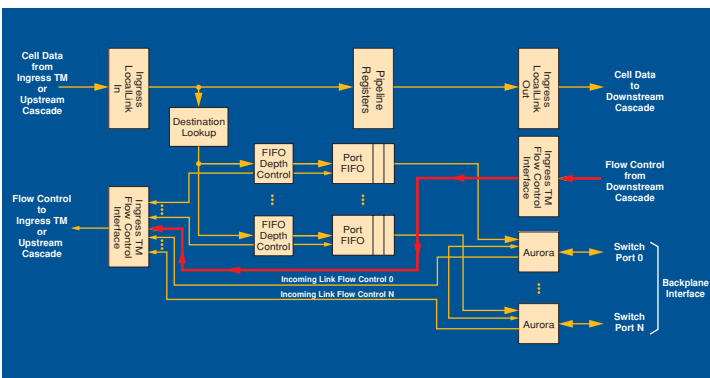


Figure 4E

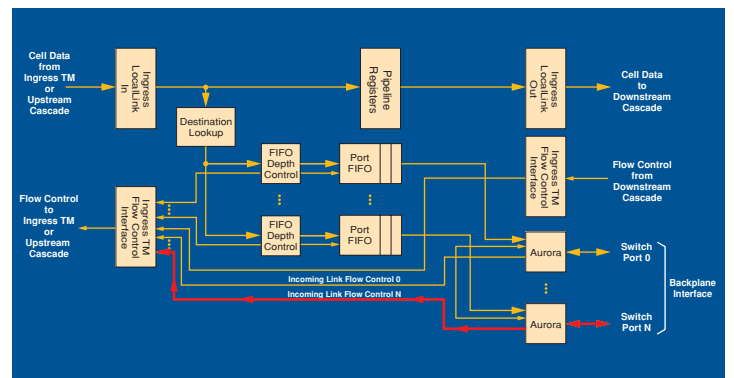


Figure 4F

Figure 4 – MFRD ingress datapath

The switch comprises four basic elements:

- The ingress datapath illustrated in the top half of the diagram
- The switch ports illustrated on the right side
- The egress datapath along the bottom
- The management interface on the left.

Also clearly visible is the use of LocalLink and the DCR bus as the interface standards in the architecture, as well as side-band signaling for flow control status on the cascade interfaces.

Figure 4 illustrates how data flows through the ingress datapath. Dataflow through the MFRD begins at the LocalLink ingress port at the top right side of Figure 4A. Incoming cells are simultaneously vectored to destination lookup and cascaded through the switch to any downstream devices in the configuration. This approach ensures efficient handling of broadcast and multicast traffic which traverse multiple devices.

In Figure 4B, destination lookup forwards the cell to the appropriate port (or multiple ports in the case of multicast or

broadcast). On this path we first enter a FIFO depth control block, which is responsible for ingress flow control for this port. If this cell triggers a FIFO event entering the buffer immediately downstream, the logic generates port-specific backpressure to the ingress traffic manager over the cascade interface (Figure 4C). This logic does not exercise flow control. It merely signals the need for flow control as the packet is forwarded to the port, illustrated in Figure 4D.

Figure 4E shows how the cascade interface also aggregates port-specific backpres-

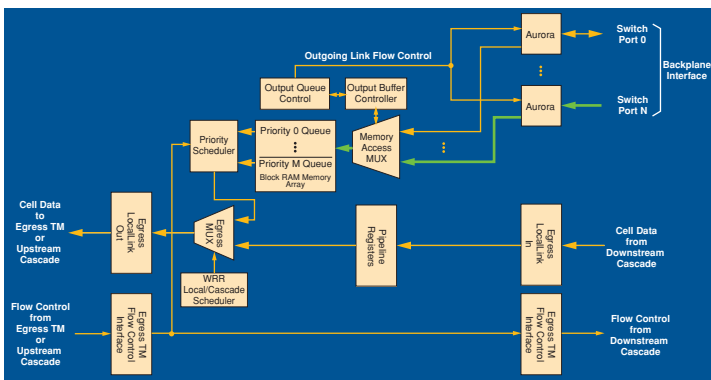


Figure 5A

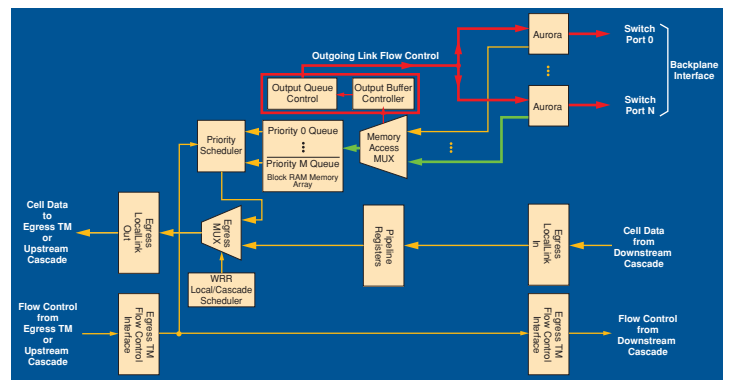


Figure 5B

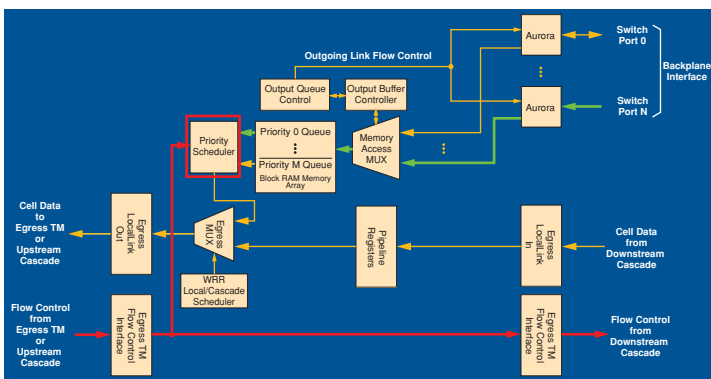


Figure 5C

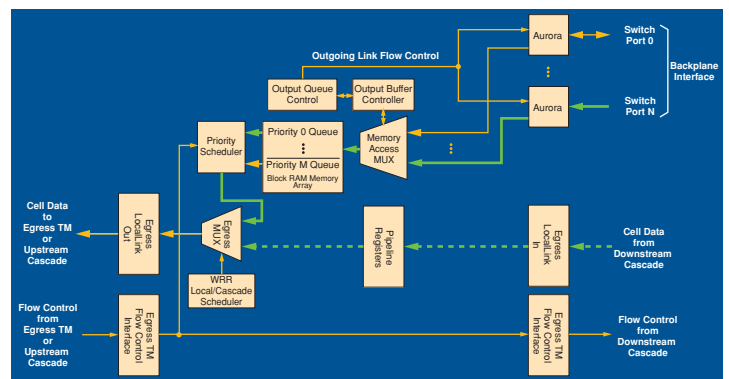


Figure 5D

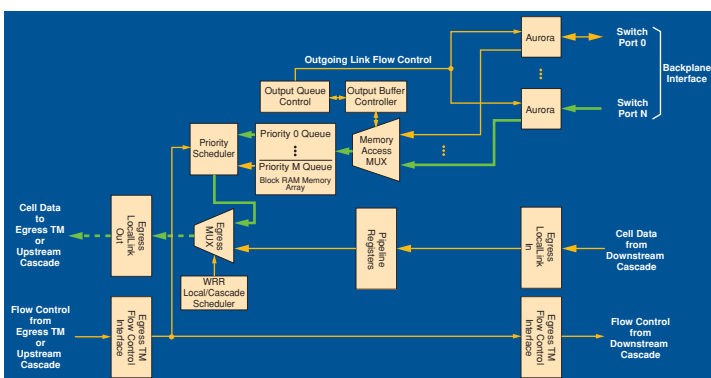


Figure 5E

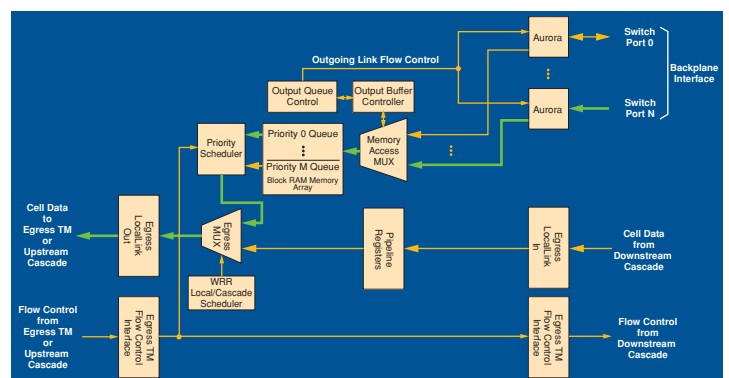


Figure 5F

Figure 5 – MFRD egress datapath

sure from downstream devices in the cascade chain, communicating flow control requirements for all ports to the ingress traffic manager. Figure 4F indicates that the architecture also supports the communication of flow control from the egress side of the switch across the serial links. This mechanism is able to refine backpressure to the ingress traffic manager with priority-specific information per port.

The egress datapath of MFRD is illustrated in Figure 5. Egress begins with the arrival of a cell at the switch port (Figure 5A). Immediately upon arrival, it is fed into a memory access multiplexer that places it into the appropriate priority queue. As shown in Figure 5B, this activity includes the generation of flow control messaging back to *all link partners* on the ingress side of the switch should this action trigger a buffer event in the target queue. This action communicates port- and priority-specific backpressure to all ingress traffic managers.

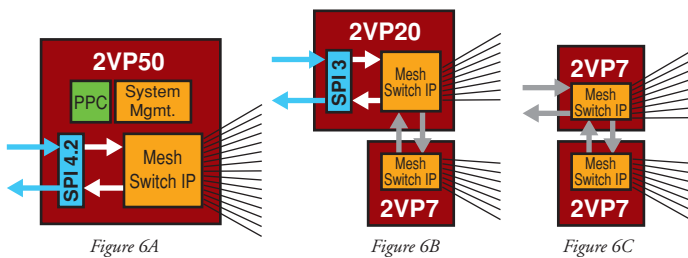


Figure 6 – Design flexibility with the MFRD

Egress from the priority queues is controlled by the priority scheduler (Figure 5C). This block can be configured using either a strict priority or weighted round robin scheduling algorithm. The scheduler is tied into backpressure from the egress cascade interface, enabling the egress traffic manager to assert priority-based flow control on the scheduling algorithm. This ensures that the scheduler will not select a priority candidate that the egress traffic manager is not prepared to accept.

Once the scheduler selects a candidate cell for egress, it is forwarded to an egress multiplexer on the egress cascade interface (Figure 5D). This block is also responsible for forwarding traffic from downstream cascade devices and must therefore ensure fair access to egress bandwidth. This is achieved

using weighted round robin scheduling through the egress multiplexer. Figures 5E and 5F illustrate how competing traffic is serialized through this mechanism.

Use Models

We have shown that the MFRD enables a great deal of flexibility to optimize the mesh switch implementation when designing your system. To illustrate this, consider the three configurations in Figure 6.

All three configurations support a 16-slot full mesh fabric. Figure 6A shows a fully integrated single-chip mesh fabric controller implementing a 10 Gb SPI4.2 interface to the application logic, a 15-port MFRD configuration, as well as processor IP suitable for implementing blade and even fully distributed shelf system management.

Figure 6B is a reduced-cost configuration of two devices that might be more suitable for supporting a 2.5 Gb SPI3-based application. Figure 6C illustrates a very low-cost

solution for applications that would use the LocalLink cascade interface from another FPGA in the Virtex-II™ and Virtex-II Pro families – a very effective way to enhance an existing system architecture.

The Xilinx ATCA Development Platform

To facilitate mesh fabric development, Xilinx has also created a full mesh reference board for ATCA, a serial backplane standard developed by the PCI Industrial Computer Manufacturers Group (PICMG™). The ATCA Development Platform is an ideal prototyping ecosystem for mesh fabric systems (Figure 7).

The ATCA Development Platform features a Virtex-II Pro FPGA with 16 integrated MGTs, 4.2 Mb of block RAM, 53,000 cells of programmable logic, and embedded PowerPC 405 microprocessors. The card is routed as a 1X full mesh and includes IP for instantiating an MFRD demo configuration. IP for instantiating a PowerPC management complex and Linux board support

package (BSP) is also available.

Programmable I/O suitable for SPI4.2, CSIX, or other interfaces is routed to personality module headers where you can integrate application-specific designs. The board also provides access to the ATCA update port and a rear transition module should your design require them.

Finally, the board features a Network Equipment Builders Specification (NEBS)-quality, dual feed, ATCA power subsystem delivering 30W to the base board and 170W to the personality module and rear transition module.



Figure 7 – The Xilinx ATCA Development Platform

Conclusion

Switch fabrics are the backbone of modern high-performance system architectures; MGT-based serial communications technology makes the benefits of mesh fabric configurations extremely accessible. With the introduction of the Virtex-II Pro Platform FPGA, Xilinx created a foundation for building such systems entirely with programmable logic. Now, with the availability of the Mesh Fabric Reference Design and ATCA Development Platform, Xilinx is making it even easier to exploit these developments and turbocharge your architectures. ●●●

For more information on these topics, please refer to the following resources:

- www.xilinx.com/esp/networks_telecom/optical/xlnx_net/mfrd.htm
- www.xilinx.com/esp/networks_telecom/optical/xlnx_net/atca_dev.htm
- www.picmg.org/newinitiative.stm