New Flexbus-4 Core

A Seamless Solution for 10 Gbps Networking Applications

by Paul Morrison
System Logic and Networking Engineer
paul.morrison@xilinx.com

Xilinx has developed a new LogiCORE™ core to give you a fully compliant solution for Flexbus-4, which is easily and quickly integrated into your networking system. Through user-configurable options, the Xilinx Flexbus-4 core provides maximum flexibility while seamlessly operating with the Applied Micro Circuits Corporation (AMCC) Application Specific Standard Products (ASSPs) to guarantee maximum bandwidth data transfers. The Flexbus-4 core is also fully compliant with the Optical Internetworking Forum's System Packet Interface-4 (OIF SPI-4) Phase I standard, a subset of Flexbus-4.

The Flexbus-4 core interfaces between Physical (PHY) and Data Link Layer devices within networking applications. The core communicates between devices using the Flexbus-4 interface standard, transferring data in excess of 10 Gbps, assuring compliance to the OC-192 data transfer standard.

AMCC developed Flexbus-4 as an interface to transfer data at 10 Gbps, the current leading-edge networking bandwidth. They have designed three different ASSPs (Ganges-I, Ganges-II, and Khatanga), also known as framer chips, that implement the PHY side of the interface. The Flexbus-4 core has been verified in hardware to operate with each of these PHY devices.

Core Functional Overview

Figure 1 shows the major blocks of the Flexbus-4 core. The source and sink Flexbus-4 blocks transfer data to and from the PHY device. The back-end interface enables transmitting and receiving data through a basic FIFO interface, and it's optimized for maximum flexibility and bandwidth. Finally, the scheduler gives you control over the sequence in which data is transferred out of the Flexbus-4 source interface.

Flexbus-4 Interface

The Flexbus-4 core communicates with the AMCC framer chips via the Flexbus-4 interface. To be compliant with the Flexbus-4 standard, the core supports the following:

- Data transfer on a 64-bit bus operating at 200 MHz, to transfer up to 12.8 Gbps.
- Symmetric interface on both the PHY and Link Layer devices, allowing the core to be used on either the PHY or the Link side of the bus.
- Point-to-point connection between a single PHY Layer device and a single Link Layer device.
- Source-synchronous clocking, where the source of the data provides a clock. This is used to simplify printed circuit board (PCB) design, by eliminating the need for complex clocking schemes.
- Out-of-band control signals, including Channel Address, Start and End of Packet, and Packet Error indications. The implementation of individual signals reduces overhead since they are not transmitted on the data bus.
- HSTL Class I I/O(EIA/JEDEC Standard EIA/JESD8-6), operating at either 1.5V or 1.8V.
- Multiple data transfer modes, including:

  Packet-Over-SONET (POS) – POS mode transmits data formatted in variable-length packets

  Asynchronous Transfer Mode (ATM) – ATM mode transmits data in 53-byte ATM cells.

  Direct-Data Mapped Mode (DDM) – DDM mode is a test mode specified by the Flexbus-4 specification, and is not supported by the OIF SPI-4 Phase I specification.

The Flexbus-4 interface provides you with configuration inputs to select between the three different data transfer modes independently for each channel. Transfer mode for each channel may be switched in-system, allowing real-time reconfiguration and the ability to operate in any networking system.

The Flexbus-4 interface also performs error checking across the data received from the AMCC framer chips. In ATM mode, if the cell received is not 53 bytes long then a cell length error is reported. Parity is checked in all modes, and an error is reported if parity is not correct.

FIFO Interface

The sink FIFO read interface is a standard FIFO interface that transmits data that has been received from the AMCC framer chips.

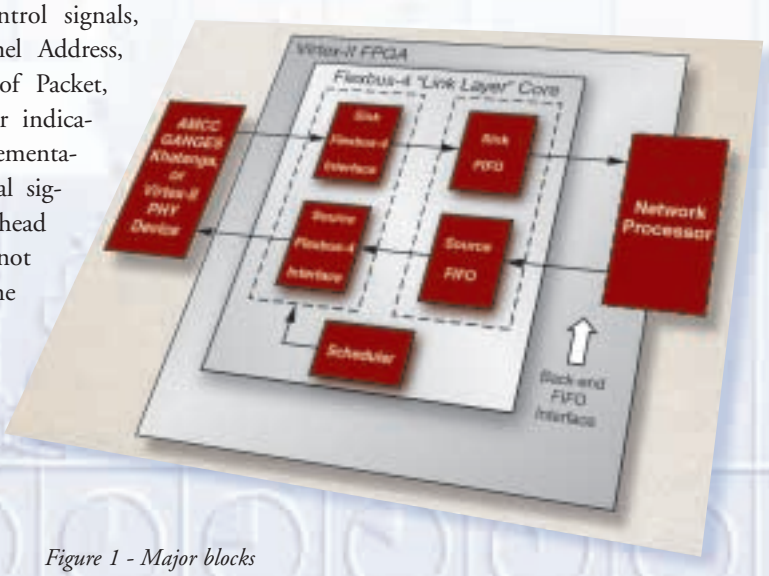

Figure 1 - Major blocks of the Flexbus-4 core

It has been optimized to support the following two options.

- Provide data for any given channel in any given clock cycle. This provides you with the flexibility to determine what channel gets serviced at what time, to ensure that the FIFO for a given channel doesn't overflow.
- Minimize overhead when switching between channels. To keep up with the Flexbus-4 sink interface, the FIFO interface needs to transfer data out as quickly as it is received. The sink FIFO read interface allows for switching between channels with no overhead, to meet this requirement.

The source FIFO write interface accepts data that is to be transferred to the AMCC framer chips. Data written into the source FIFO can target any channel on any clock cycle, providing flexibility in meeting bandwidth requirements for each channel within the system. Additionally, each channel has its own source FIFO, such that if the FIFO for one channel gets full, data transfers can still continue to all other channels.

Scheduler

The scheduler is a programmable sequence table, designed to specify the channel order that the Flexbus-4 source interface uses to send data out onto the Flexbus-4 bus. The scheduler is only used in the source direction of data transmission, and does not effect the data flow in the sink direction. The flexibility of the scheduler is extremely useful for meeting bandwidth requirements for different channels, as well as using available bandwidth efficiently.

Figure 2 shows an example of the programming of the scheduler, and how it translates to data transferred across the Flexbus-4 bus. The transfer rate per channel provided by the core is a ratio of the number of times a particular channel is written into the scheduler vs. the total number of locations used in the scheduler. If a particular channel requires four times more bandwidth than other channels (as is shown in Figure 2), it is written into the scheduler four times more often. Since the order and frequency of channels written into the scheduler is completely user-dependent, you are able to get any combination of transfer rates between channels that you require.

To take advantage of the maximum available bandwidth of 12.8 Gbps, the scheduler ignores channels that do not have data available to transfer, and splits up the available bandwidth between the channels that are ready to transfer data. If only a single channel within the system contains data, then it will get the full bandwidth with no overhead required when checking for data available on all other channels. If all of the channels within the system contain data, they will get bandwidth according to the programming of the scheduler.

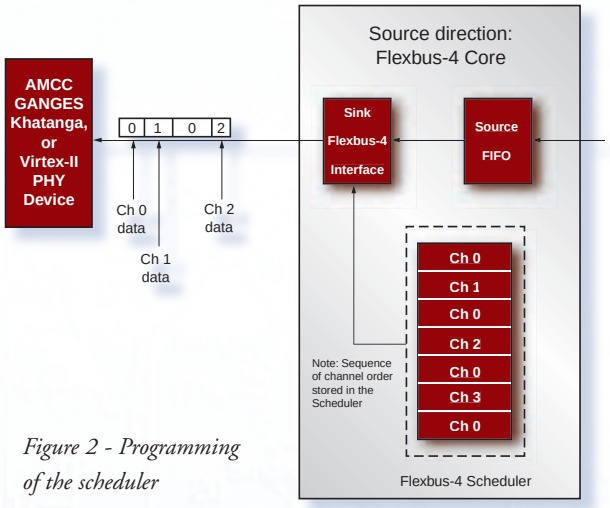

Figure 2 - Programming of the scheduler

What's Provided

To make implementation and integration quick and easy, Xilinx provides several additional support features in addition to the Flexbus-4 core. The following are included:

- **Loopback logic** – A design to loop data from the sink FIFO to the source FIFO is provided in both Verilog and VHDL. In addition to providing the loopback functionality required in many networking systems for error isolation, this design also provides an example interface that will be extremely useful when you design your interface to the core.
- **Instantiation Templates** – An instantiation template for the Flexbus-4 is provided in both VHDL and Verilog. This template provides an example of instantiating the core, as well as showing how to create source-synchronous clocking using Virtex-II Double-Data Rate (DDR) registers.
- **Build Script** – A push-button build script is provided that places and routes the Xilinx core and the loopback logic. This is useful for demonstrating that the core meets the 200 MHz timing requirement within the Virtex-II FPGA, as well as providing a sample build script for the core.
- **Demonstration Testbench** – A demonstration testbench to show toggling of the signals interfacing to the Flexbus-4 core is provided in both VHDL and Verilog. This can be easily modified to view operation of the Flexbus-4 core, and the different core configuration options available.

Conclusion

The Xilinx Flexbus-4 cores are currently available from the Xilinx IP Center at *www.xilinx.com/ipcenter/flexbus/flx4.htm*. There are three cores currently available, designed to interface to network systems that contain a single OC-192 channel, four OC-48 channels, or sixteen OC-12 channels. Each of these cores has been verified in hardware to operate with the AMCC Ganges-I, Ganges-II, and Khatanga framer chips at the OC-192 transfer rates.

Beyond being compliant with the Flexbus-4 specification, this core has been optimized to provide maximum flexibility, to ensure optimal bandwidth for your system's configuration. The Xilinx Flexbus-4 solution also enables you to quickly develop an OC-192 interface ensuring you the fastest time to market.