Migrating to Virtex-5 – without falling off a cliff

By Andrew Reddig

Upgrading technology always involves some development and integration risk. The use of well-defined IP cores with high-level, consistent interfaces minimizes the risk of technology insertion and allows end users to successfully migrate to new technology.

FPGA technology is becoming more and more like software. When a new processor is released, users will often need to recompile and rebuild their applications to make use of newer hardware and take advantage of the performance gains of the new processor. However, it is rare that the user has to rewrite their application or rearchitect their entire system just to upgrade the processor.

With careful selection of the right abstraction boundary and a comprehensive Developer's Kit (DK), board and system vendors can now offer a similar degree of investment protection when FPGA technology is being used.

A large number of systems deployed today implement signal processing using Xilinx Virtex-II Pro FPGA technology. The Virtex-5 family offers significant improvements in density, power, and speed, all of which are critical for both current and future applications. Migration of existing user applications – both software and firmware – to next-generation products using Virtex-5 will extend the capability and service life of existing systems and create better solutions for end users, as long as the migration can be successfully managed.

Background

A recent trend in high-performance embedded computing is to use reconfigurable processing technology to achieve ever-higher performance within demanding space, power, and volume constraints. By using FPGAs, board and system vendors can offer COTS products that can nevertheless be precisely tailored, in both firmware and software, to the end user's application. This maintains the COTS business model of amortizing development costs across multiple customers and programs even though the hardware itself is now essentially customizable.

But one of the challenges of COTS in general, and FPGA technology in particular, is the rate of change of the underlying technology. The major FPGA vendors are all competing in a rapidly changing market, and new FPGA devices are being developed every 18-24 months. Since the driving force behind insertion of FPGA technology is raw performance, users have an obvious incentive to migrate rapidly to newer technology when it becomes available. At the same time, deployed systems comprise a substantial investment in architecture, application design, software, qualification testing, and integration that needs to be leveraged for new technology to be cost-effective.

From a user perspective, a COTS FPGA platform provides several things:

- A hardware product that contains one or more FPGA processing elements
- Memory and other onboard resources for the FPGA to use
- Connectivity between the FPGA and external devices, either through a backplane interface such as VITA 41 (VXS) or through external I/O such as fiber optic transceivers
- A developer's kit with Intellectual Property (IP) cores that abstract the interface between the user's firmware and the resources provided by the vendor

To support migration from product to product, whether from PMC modules to XMC modules, from VXS to VITA 46 (VPX), or from Xilinx Virtex-II Pro to Virtex-5, the definition of the interfaces and the flexibility of the IP cores provided in the DK are the most critical factors. If the DK performs well, the underlying hardware can improve with technology, but the user's investment in their application IP and software is protected through the interfaces provided by the DK.

Each DK will include several different IP cores to support the various features of the underlying hardware. For example, the DKs for Tekmicro's VXS products typically contain IP cores for the following functions:

- A/D interfaces
- Memory controllers (DDR, SRAM)
- QuixStart bitstream management
- QuixStream interconnect, using either Aurora, GbE, or VITA 55 Virtual Streaming Protocol
- JazzStore SoC data recorder module (as an optional component)

Each of the above IP cores presents an abstract interface to the hardware that supports migration from existing Virtex-II Pro based products to future Virtex-5 platforms with minimal application changes. The following discussion will focus primarily on the JazzStore System-on-Chip (SoC) core as an example, but the discussion applies equally to all of these functions.

Data recording core

The JazzStore SoC core integrates several elements of the FPGA device to create an SoC data recorder capability contained entirely within the FPGA device. A block diagram of the JazzStore SoC core is shown in Figure 1.

While each element of the core will benefit from the improved capability of Virtex-5 FPGA devices, straightforward migration of a user design requires that the core designer isolates the user from the internal implementation. By maintaining the same interface to the user application, the core provides a "black box" capability that encapsulates the changes between Virtex-II Pro and Virtex-5. This allows the user to migrate their design to newer technology without design or architecture changes.

First, we will describe the functions performed by each element of the core, and then we will review how each element benefits by migration to Virtex-5.

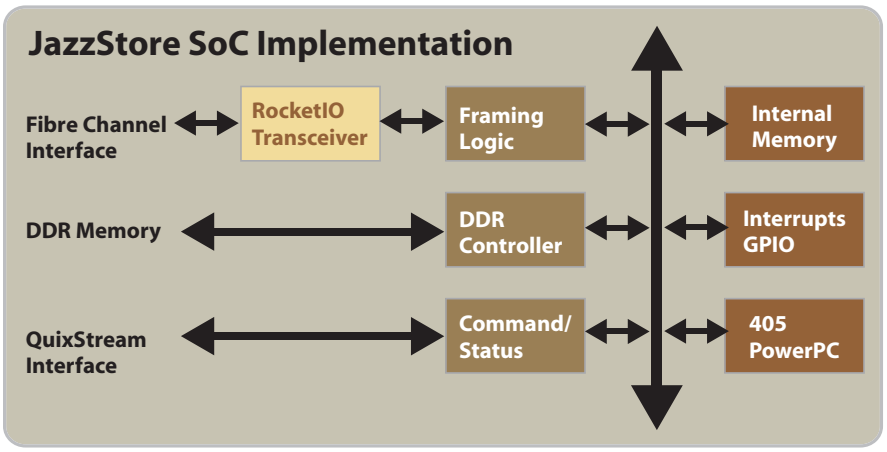

Figure 1

The external interface uses a Xilinx RocketIO SERDES to implement a 2.125 Gbps Fibre Channel interface. The RocketIO module offers the serializer-deserializer, 8B/10B encoder, and phase locked loop clock circuit to implement the Fibre Channel FC.1 interface between the core and the external fiber optic transceiver.

The core then uses dedicated logic to implement the FC.2 state machines and framing to stream data between the on-chip 405 PowerPC processor and the external interface. The software running in the PowerPC implements the appropriate higher-level protocols along with file and session management and the Windows-compatible FAT32 file system. The core reuses the DDR memory controller from the DK to provide buffering of the data stream and of file system data structures.

To the user, the core appears as a simple FIFO interface with a separate control port to manage record sessions and other out-of-band control and status activity. While the internals of the core utilize the RocketIO and PowerPC resources, from the user perspective the implementation simply streams data to disk. The type of disk (single drive, RAID, or JBOD) and the management of the file system are all handled within the core. This provides a level of abstraction that supports the goal of easy migration from Virtex-II Pro to Virtex-5 without changing the user application.

When the JazzStore SoC core is migrated to future Virtex-5 based VXS products, the internal implementation could change in several ways:

- The PowerPC will run at a faster clock speed, allowing each PowerPC to support faster disk interfaces and potentially be multiplexed for additional instances of the data recorder core.
- The DDR memory will change to DDR2 and run at a faster clock speed. These changes will be encapsulated within the DDR memory controller itself and won't affect the JazzStore SoC core directly, but will support higher-memory bandwidth.
- The Fibre Channel interface will migrate to 4.25 Gbps or higher, using faster SERDES technology as it becomes available.
- The higher-density FPGA device will allow more data recorder cores to be instantiated within a single device, increasing the channel density and overall throughput.
- The FIFO interface to the user's firmware will use faster on-chip memory and will run at a higher clock rate.

Even with all of these changes, the VHDL interface to the core will be essentially unchanged. Many applications will be transportable from Virtex-II Pro to Virtex-5 through the use of new IP cores and a simple recompile of the user application with a new DK.

Interprocessor communications core

A critical piece of the design of an FPGA-based signal processing system is the overall data flow architecture and the method used for communications between different processing nodes in the system. If the communication protocol is sufficiently abstract, the migration from Virtex-II Pro to Virtex-5 becomes purely an implementation problem inside the core, without affecting the user's software or firmware. As with the data recording core, encapsulating the implementation within a black-box core significantly reduces the risk of migrating existing applications to Virtex-5.

Interprocessor communications are implemented in the DK using the QuixStream core, which supports communication either between FPGA nodes or between FPGAs and external systems. The core provides a common "socket"-like interface between the internal FPGA application and an outside device. The most common choice for the underlying transport is GbE, but the interface can be implemented using a wide range of transport technologies.

The user application typically connects the control and status functions of the user IP to the interface and then implements a network connection to the external control processor. A block diagram of a workstation-to-FPGA connection using QuixStream and GbE is shown in Figure 2.

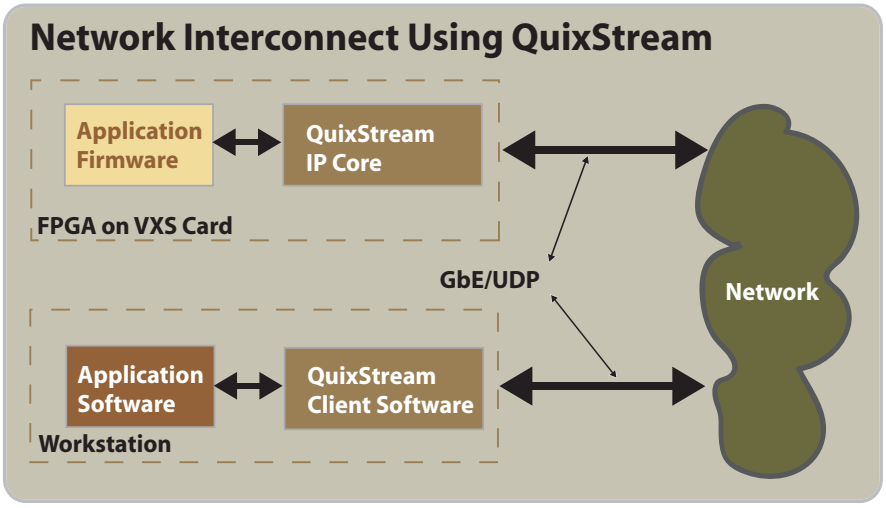

Figure 2

Because the interface uses an open protocol as the underlying transport, architecture migration from Virtex-II Pro to Virtex-5 is extremely straightforward. The workstation side of the interface simply communicates to a UDP socket through a network connection and has no coupling with the FPGA implementation technology. Within the FPGA, the Virtex-II Pro implementation uses dedicated logic to implement GbE and UDP processing. The Virtex-5 implementation could take advantage of the embedded device resources to implement a GbE Media Access Control (MAC), reducing the required resources and also increasing the flexibility of the interface, but again without changing the user's application.

Low-risk migration

Technology upgrades are a necessity to extend both the functionality and life cycle of critical embedded systems. A comprehensive set of IP cores with well-defined boundaries has the potential to dramatically reduce the development and integration risk of upgrading systems to newer FPGAs such as Xilinx's Virtex-5 family.

Given the long life cycles of deployed, high-performance embedded systems and the relatively high investment in integration and qualification, continued acceptance of FPGA technology depends on this promise becoming a reality. The migration to Virtex-5 will shortly provide demonstrations of this paradigm in real-world applications. CS

Andrew Reddig *is President/CTO and cofounder of TEK Microsystems, Inc. TEK Microsystems was founded in 1981 as an engineering services organization while Andrew was a student at M.I.T. Today, TEK Microsystems is the leading supplier of advanced streaming I/O products and provides embedded VME solutions employing a combination of onboard fabric, FPGA, and PowerPC technology.*

To learn more, contact Andrew at:

TEK Microsystems, Inc.
2 Elizabeth Drive
Chelmsford, MA 01824
978-244-9200
areddig@tekmicro.com
www.tekmicro.com

Signal processing

Signal processing core libraries

Ultimately, speed is measured by the clock rate at which a device operates and the number of clocks from input to output for a given function. The clock rate is driven directly by FPGA technology, with Virtex-5 offering clock rates approximately twice as fast as Virtex-II Pro. For a given implementation, this will result in a linear improvement in performance.

Migration of user applications from Virtex-II Pro to Virtex-5 is ultimately driven by the performance of the signal processing functions being undertaken within the FPGA. The use of IP cores with well-defined boundaries helps make migration successful for the "utility" functions such as memory, I/O, recording, and communications. Signal processing functions can also benefit from the same techniques.

In addition to the IP cores that provide the bridge between the user application and external interfaces, applications typically make use of IP cores for specific signal processing functions such as Fast Fourier Transforms (FFTs) or Finite Impulse Response (FIR) filters. These IP cores are analogous to library calls in the software domain, providing a high-level interface between the application and the underlying processing hardware.

For high-performance embedded computing, most IP cores follow a streaming I/O model, providing FIFO-based input and output streams with a separate control port. Sensor data is presented to the input FIFO and clocked into the core; after a given number of clock cycles, results are available at the output FIFO. Typically, the width of the FIFO is selectable at compile time, and often the signal processing parameters (FFT window size, for example) are dynamically configurable at runtime to support adaptive algorithms.

The internal implementation of an FFT core can be optimized for speed or for resources, with tradeoffs between the two. Higher speed is typically achieved through more parallelization, which consumes more resources.

Parallelization is driven by how many resources the function consumes relative to how many resources are available in the device. Virtex-5 offers advantages in the internal implementation through the use of higher-performance DSP slices and multiply-accumulate primitives. For some functions, these embedded device features can be used instead of synthesized logic, reducing the number of logic slices needed for a given level of parallelization.

Finally, Virtex-5 devices offer higher gate counts per device, and perhaps more importantly, higher density for a given power limit. Between the overall density improvement and the use of embedded features, a given implementation can either maintain the same degree of parallelism – and fit more functions into the chip – or increase the parallelism and achieve higher performance.

From the user's perspective, the end result of all of the above tradeoffs is simply that the internal performance of the FFT core is faster, with more tradeoffs available between speed and resources. The semantics of the interface to the FFT core, and therefore the use of the core by the user's firmware, do not change from device to device. This enables the user to easily migrate their internal firmware from Virtex-II Pro to Virtex-5 without redeveloping other modules.