

NATIONAL INSTRUMENTS: DEVELOP AND DEPLOY FPGAs EASILY

1 October 2008: Develop and Deploy FPGAs EasilyINTERVIEWEE. RICK KUHLMAN - PRODUCT MANAGER, *LabVIEW* FPGA

TEL. (512) 683-6806

EMAIL. rick.kuhlman@ni.com

COMPANY. NATIONAL INSTRUMENTS

WEB. <http://www.ni.com/>**Q. First of all, tell us a little bit about yourself and your responsibilities at National Instruments (NI).**

A. My name is Rick Kuhlman and I am the product manager for NI *LabVIEW* FPGA. I earned a bachelor's and master's degree in electrical engineering from the University of Tennessee as well as an MBA. After interning with National Instruments in 2001, I joined the NI Engineering Leadership Program as a full-time applications engineer. Currently, as a product manager I am responsible for *LabVIEW* FPGA technical and business objectives and work closely with NI R&D and NI customers to determine key product features, develop product strategy, and drive inbound and outbound initiatives surrounding *LabVIEW* FPGA.

Q. Philosophically, your CEO, Dr. James Truchard, has communicated his excitement about FPGAs and how NI can help developers deploy FPGA-based computing quickly and easily. Why is Dr. T so excited about FPGAs?

A. Dr. T's excitement stems from his belief in the high-level message of the company. Dr. T is reinventing how scientists and engineers create test, industrial, and embedded systems – coined graphical system design. He sees field-programmable gate array (FPGA) technology as a key solution for improving system performance, cost, and reliability, not to mention making impossible objectives feasible. Unfortunately, programming FPGAs has historically been difficult and relegated only to digital design engineers. Dr. T's vision is to expand beyond the traditional bounds of this valuable technology in two directions:

(1) Deliver current digital design engineers a shorter time to market, more intuitive tools, and rapid prototyping capabilities.

(2) Enable the domain expert engineer or scientist, who would never have been able to use the technology otherwise, to program FPGAs.

Q. Tell us specifically about *LabVIEW* FPGA 8.6. For those new to NI, what is *LabVIEW*, and what features does it have specifically to make FPGA-based designs easier for developers?

A. *LabVIEW* in general is a graphical dataflow programming language where, instead of text, users specify the execution of their program with icons and wires. It is much like creating an engineering diagram where icons represent functions; wires carry data; and structures, such as looping and cases, are special regions of the code. Using this high-level approach, programmers can avoid difficult details like memory allocation and syntax, focusing rather on their algorithm and core competency. At the same time, *LabVIEW* still allows programmers to dig down to low levels and optimize with things like in-place memory allocations, inline C-nodes, and execution trace debugging.

NI has extended this *LabVIEW* paradigm of programming from desktop and real-time systems to the FPGA. Perhaps the most unique characteristic of *LabVIEW* and its usage with FPGAs is the ability of *LabVIEW* to easily represent parallelism, which is a key differentiation between programming for processor-based systems and FPGAs. When a user creates two parallel paths of data in *LabVIEW* FPGA, they physically become parallel processes in the hardware. *LabVIEW* FPGA also has many built-in functions that make it easy to get up and running quickly including DMA data streaming, on-board memory communication, signal processing, control, digital communication buses, and more. Couple *LabVIEW* FPGA with NI off-the-shelf plug-in boards or FPGA-enabled embedded systems and users have drag-and-drop I/O and signal conditioning enabling extremely quick times to first prototype.

Q. One of the features new in 8.6 relates to “Intellectual Property” or “IP.” Tell us about how *LabVIEW* 8.6 works with IP, and especially about how it hopes to make it easy to import external IP and manipulate IP for the design.

- A. IP in *LabVIEW* FPGA can come from a variety of sources. First, it can come directly from the *LabVIEW* palettes. The IP blocks available in *LabVIEW* FPGA out of the box includes basic math, signal processing, control, data transfer, storage, I/O interfaces, and features to help users write their own IP. Second, users will find valuable IP from our IP generators like the NI filter design toolkits, which can generate code for specific digital filters including Butterworth, Elliptical, IIR, FIR, multirate, and even adaptive filters. Finally, *LabVIEW* FPGA has ways to integrate any external IP written in other hardware description languages (HDLs).

Using the new *LabVIEW* 8.6 feature called Component-Level IP (CLIP), users can acquire IP from any source, define interfaces between *LabVIEW* FPGA and the port map of the IP, and smoothly use the external IP in the *LabVIEW* FPGA context. This opens the doors to IP vendors, HDL-savvy, and open-source code blocks to be used in *LabVIEW* FPGA and subsequently on NI FPGA-based hardware platforms.

Q. A few years back, NI introduced *CompactRIO*. Many FPGA developers develop for FPGAs from Xilinx, Altera, and the like; others buy FPGA-based boards from vendors like Pentek or Vmetro. What is *CompactRIO*? What market segment does *CompactRIO* target? How is it different from other FPGA solutions?

- A. NI *CompactRIO* is a high-level platform for developing real-time control and acquisition applications. *CompactRIO* is composed of a real-time PowerPC processor coupled with a reconfigurable Xilinx FPGA that is directly connected to a wide range of modular I/O modules. Both the real-time processor and FPGA are programmed directly with *LabVIEW*. Engineers and scientists deploy *CompactRIO* in a wide range of circumstances ranging from the desktop and lab benches to field deployed embedded machines. Because of the tight integration of hardware and software components achieved between *CompactRIO* and *LabVIEW*, domain experts who normally would not have the technical expertise to develop a complicated embedded system can take advantage of the determinism and reliability associated with a real-time computing platform and the customization, flexibility, and speed of the FPGA. Using *LabVIEW* as the graphical system design tool for the entire application empowers developers to shorten time to market and focus on their end application features and implementation, rather than on drivers, boot loaders, thread schedulers, and the other common woes of traditional embedded designers.

Q. At this year’s NI Week your company announced, *Single-board RIO*. Help us place this in the company strategy vis-a-vis *CompactRIO* and *LabVIEW*.

- A. NI Single-Board RIO delivers a smaller form factor, a single-board option for custom enclosures, and is the next step in lowering the cost of deploying the reconfigurable I/O (RIO) architecture in high-volume applications. NI Single-Board RIO reuses the *CompactRIO* architecture of real-time processor + reconfigurable FPGA + I/O, but cost optimizes the system by removing the metal enclosure integrating all three primary components on a single printed circuit board (PCB). NI Single-Board RIO is targeted at customers who see the flexibility, ease-of-use, and time-to-market benefits of *CompactRIO*, but need to cost optimize for a high-volume deployment or need to build the RIO architecture into a smaller custom enclosure. Because NI Single-Board RIO shares the same architecture as *CompactRIO*, designers can prototype a system using the modular *CompactRIO* platform while they define their I/O requirements and algorithms. When they are ready to deploy to higher volumes, they can reuse their *LabVIEW* code directly on NI Single-Board RIO – in many cases with no code changes. For more information on using *CompactRIO* as a rapid prototyping platform, check out the links in the last section of the interview.

Q. Wow, on the FPGA front, NI certainly has a lot to talk about! Unfortunately, in the span of our short interview, we can really only touch upon highlights. Would you do our readership a favor and give us some Web URL's where they can go for more information about *LabVIEW*, *CompactRIO*, and Single-board RIO. They are always especially interested in online learning, tutorials, demos, webinars and the like.

- A. Overall *LabVIEW* FPGA home page – <http://www.ni.com/fpga>
 Introduction to *LabVIEW* FPGA webcast – <http://zone.ni.com/wv/app/doc/p/id/wv-229>
 Information on Component-Level IP (CLIP) with video demos – <http://zone.ni.com/devzone/cda/tut/p/id/7444>
 List of available IP from in-product to user-submitted functions – www.ni.com/ipnet
CompactRIO home page– <http://www.ni.com/CompactRIO/>
 NI Single-Board RIO home page – <http://www.ni.com/singleboard/>
 NI Single-Board RIO white paper - <http://zone.ni.com/devzone/cda/tut/p/id/6265>

Q. What is the future of FPGA-enabled devices from National Instruments?

- A. We will continue to develop FPGA devices along at least two vectors. First, we will continue to create FPGA-based embedded systems like the *CompactRIO* platform. In this regard, we want to create more powerful devices for higher performance and create smaller more “embeddable” targets for volume deployment. Second, we want to create more FPGA-enabled plug-in boards. Aside from extending the popular R Series FPGA-based data acquisition boards, we have also recently released the new NI *FlexRIO* platform. NI *FlexRIO* is a plug-in FPGA board with direct connection to FPGA pins on the frontend and bus interfaces, memory interfaces, and surrounding electronics on the backend. With NI *FlexRIO*, we are investing in an architecture that will enable NI as well as NI partners to use standard interfaces to create all types of acquisition, generation, signal conditioning, control, communication, or other myriad types of FPGA-based plug-in boards by simply creating the respective terminal block for the standard NI *FlexRIO* board. Additionally, any new technology will still use the *LabVIEW* FPGA platform along with the hooks to integrate any IP as the method of programming these devices.

Q. Thank you for this interview.