FPGAs supercharge instrument flexibility

Larry Desjardin  - March 25, 2013

Most modern instruments include one or more FPGAs (Field Programmable Gate Arrays) in their circuitry. Almost all discrete logic is consolidated there, and increasingly vendors are turning to FPGAs to implement DSP (Digital Signal Processing) functions rather than deploying ASICs (Application Specific Integrated Circuits) or dedicated DSP chips.

Module vendors in VXI, PXI, and AXIe often deploy FPGAs to perform the interface function to their respective backplanes, in addition to the specific instrument functions of the module. Since modular systems attain much of their speed advantage by avoiding ASCII communication found in traditional instruments, the FPGA creates a memory map interface that, along with the software driver, defines the automation interface of the instrument. Eliminating the processor found on traditional instruments places the FPGA front and center in defining the operation of the module, but often at hardware speeds.

But there is a trend afoot that may be as disruptive as modular instrumentation itself: Giving users the ability to directly customize the FPGA. While the automated test metaphor since the 1970s has been fixed-definition hardware instruments controlled through flexible programming, this recent trend now allows users to program the hardware definition of the instrument itself. That’s Flexibility with a capital F. Maybe all caps, in fact.

This is easier said than done. Vendors must isolate the critical functional blocks, also known as IP (Intellectual Property), from the user-programmable space. For example, the PCIe (PCI Express) interface and backplane trigger access must remain intact regardless of the user’s programming. The design must make sure that catastrophic events can’t occur that cause damage to the product. Finally, what programming method does the user deploy to define the custom functionality? HDLs (Hardware Description Languages) have historically been the tools of choice to define the embedded functionality. However, most test labs (often the targeted users) have few, if any, HDL developers.

With this in mind, I performed a quick survey of what is available from the marketplace. Call it the FPGA Tour.

The FPGA Tour

National Instruments is probably the most visible of the FPGA proponents. Using LabVIEW FPGA as the tool that allows users to customize the embedded FPGA, users program in LabVIEW, not HDL. Since these users may also be deploying LabVIEW as the automation language, it is a familiar programming language, and one that gives them a choice of deploying their own IP either on the host controller, or directly on the instrument’s FPGA. NI has offered FPGA programming for some time, accessed through their FlexRIO-enabled modules. These may be custom modules where a user can add their own instrumentation circuitry through a digital I/O interface. Or the FlexRIO architecture may be embedded in commercial NI digitizers or signal generators, allowing the user to...
customize the operations of those products. NI’s latest salvo in this area was the introduction of the VST (Vector Signal Transceiver). Here, the entire instrument is an open source LabVIEW virtual instrument, including the FPGAs. At its introduction at NI Week, I witnessed how a LabVIEW-knowledgeable user could click down through the VI hierarchy to customize the hardware or software operation of the instrument.

NI lists four key application areas they see user programmable FPGAs deployed: Real time continuous measurements, custom triggering and acquisition, protocol emulation, and closed-loop and real-time test. NI's vertical tool set avoids third-party tools as an explicit strategy to deliver ease of use for end users. Two good third party examples are the channel emulation IP for the VST developed by Averna, and the signal processing IP developed by Gefle Testteknik for testing small cell base stations.

ZTEC Instruments has taken a different approach with their modular instruments. While they deploy FPGAs that support some degree of customization, they found the user-customizable approach to be complicated and non-optimized for cell size and speed. Instead, they work with a customer on a project basis to incorporate the users’ IP into the module. Three applications areas requiring FPGA customization include high-energy physics deploying multi-channel acquisition and parallel processing, military radio communications where the protocol may be secret, and niche users who may have their own FPGA code. ZTEC has found the current set of FPGA tools to be powerful, but complex, and the high-level languages to add another level of inefficiency and compile time. By dealing with a customer on a project basis, they remove these complexities.

Sundance has offered FPGA-customizable products in PXI for some time. They started with a set of ADC and DAC modules, each with customizable FPGAs. Since that time they have released FPGA-based processor modules that process data coming from or going to another PXI module, or interface to a user-created carrier card through a 54 wide LVDS bus. All their customers use VHDL. Typical applications are FFTs and other DSP. These may be used for prototyping a military radar system in PXI, or for reducing and buffering streamed data before being sent to the host controller, such as the data from a high-speed camera. Sundance supports Windows and Linux environments.

Teradyne offers a PXIe-based High-Speed Subsystem (HSSub) that generates and analyzes serial buses above 50MHz. This is often used for testing of avionics, where unique serial buses are prevalent. For each specific DUT (Device Under Test), the HSSub Runtime Defined Instruments are configured from the tester. These modules each incorporate FPGAs that provide low-latency real-time interaction between the tester and the DUT. Closed-loop testing requires low latency that cannot be achieved through a software-based system. End users may provide their own FPGA code for custom applications aided by Teradyne FPGA code libraries that bind it to the HSSub infrastructure. The entire subsystem acts as an LXI instrument which then executes the serial bus algorithms at hardware speed.

Geotest-Marvin Test Systems offers two families of FPGA cards for PXI, the GX3500 and GX3700e. Each is primarily a Digital I/O card used in mil/aero and semiconductor testing. For programming the FPGAs, users typically use schematic entry or VHDL in conjunction with the FPGA vendor tools. The products can be configured with expansion boards that provide buffered I/O interfaces for the common logic families, or a video generator board. Alternatively, users may add their own custom mezzanine card that sits piggyback on the GX3500, all within the envelope of a single PXI module, to create their own custom instrument. Geotest supports Windows and Linux environments.

VTI Instruments introduced their EMX PXIe series a little more than a year ago. It is essentially
their latest series of dynamic signal analyzers aimed at noise, vibration, and harshness testing of mechanical machinery. Their **EMX-series digitizers** have on-board FPGAs, which allows local processing to unload the host. In particular, DSP algorithms may be customized to a particular application. Take the example of rotating machinery analysis. The frequencies of interest will be at the fundamental and harmonics. By customizing the DSP algorithm, a user can focus on those frequencies of interest. VTI offers **Simulink**-based graphical tools for the development of custom IP blocks. These Matlab-based tools from **MathWorks** offer an alternative to HDL-based design.

While the above focuses on PXI vendors, AXIe vendors are also offering FPGA customization. **Agilent Technologies** offers **high-speed digitizers** in PCIe, PXI, and AXIe formats that include on-board real-time signal processing using FPGAs. A good example is the **M9703A AXIe digitizer** which sports four large FPGAs behind its bank of high-speed 12-bit ADCs. Similar to ZTEC, Agilent will work with a customer to allow that customer to incorporate their own IP or custom algorithms on a case-by-case basis. Agilent's prime concern is keeping the measurement integrity of the instrument, and has chosen this collaborative process in order to do so. Applications for user IP include a wide array of OEM applications, mil/aero applications, and high-energy physics and data acquisition.

**Guzik Test and Measurement** offers the **ADC 6000 series** AXIe digitizer modules. This series is essentially a 40Gs/s AXIe digitizer that may be configured as a single channel, two 20Gs/s channels, or four 10Gs/s channels on a single AXIe blade, all offering deep memory and an array of FPGAs for back-end processing. Customization of the FPGAs is done by Guzik after technical consultation with the customer. Guzik’s latest module, the **DP 6000 Digital Processor Module**, is a companion module with two large FPGAs, a 40GB/s local bus, and 25.6 GB/s PCIe bus for offline storage and streaming. This module’s FPGA is completely user configurable using a firmware development kit. Guzik supplies Verilog code for the basic functions of data transfers, memory control, and so forth. Applications utilizing FPGA customization include high-energy physics, mil/aero radar simulation, or non-test applications involving real-time acceleration of simulations and databases.

**Observations**

Looking at all of these together, a few patterns emerge. While NI and VTI utilize LabVIEW FPGA and Simulink respectively, therefore eliminating the need for HDL programming, the other vendors offer tools that assume some degree of HDL expertise. There is some debate about the ease of use of such tools versus the gate count and size efficiency, and different segments may have different preferences. Nevertheless, my hunch is for FPGA programming to become widespread, non-HDL programming will be required. The advent of **OpenCL** (Open Computing Language) may offer another method of doing so.

The second observation is that each vendor implements user customization differently. Each offers a specific tool hierarchy, often based on the tools of the FPGA vendor, and there is little or no portability to another vendor’s FPGA. In the embryonic stage of this technology this may be entirely acceptable. Let’s see what works and doesn’t through a dozen experiments. But in the long run users are going to desire some degree of standardization and portability, as they have with SCPI or IVI.

My third observation is that all user customized FPGAs exist in modular formats, like PXI or AXIe. I was unsuccessful in finding any traditional bench instruments that support FPGA customization. Maybe there are, but I didn’t find any. This is interesting, because there is nothing inherent in FPGA customization that is uniquely modular.
My final observation regards the power unleashed by giving users this option. User programmable FPGAs in instruments deliver as large of a leap in customization as did the first programmable instruments in the 1970s. Whether a user is defining unique pattern and level triggers, a custom bus interface, or inserting a proprietary DSP algorithm, they are doing so at hardware speeds. My modular prediction #3 (I Can’t Drive 55) forecasted significant speed breakthroughs for 2013. User programmable FPGAs deliver that, and may prove to be as large of a paradigm changer to the test automation marketplace as modular instruments themselves.