Through a recent collaboration, Geotest–Marvin Test Systems and JTAG Technologies integrated one of JTAG’s boundary-scan controllers into Geotest’s preconfigured PXI test system for the production floor. This suggests that PXI may be moving into new realms in manufacturing test. I spoke with Mike Dewey, Geotest’s senior product marketing manager, to learn more about how this application area is evolving.

Q: What prompted the collaboration?
A: We have been seeing more interest in using boundary scan for production testing as well as for programming of flash memory, CPLDs, and the like on the production floor. JTAG Technologies had PXI boundary-scan products and extensive support software, which made it easy for the companies to integrate their technologies into a preconfigured system.

Q: But hasn’t boundary scan been available on PXI for a long time?
A: Yes, boundary-scan controllers have been available almost from PXI’s inception [in 1997]. But the technology has been mostly used as an adjunct to in-circuit structural testing or as a separate test methodology and not part of a functional-test methodology that PXI systems typically provide. For whatever reasons, combining the two test techniques hasn’t caught on, particularly in the North American marketplace, until now.

Q: Why is combined test catching on now?
A: What has happened is a loss of access to signals on boards and modules due to shrinking feature sizes, buried vias, and the like. This has made bed-of-nails probing or in-circuit testing more difficult, so the industry has been moving away from using in-circuit test systems and begun using x-ray and optical inspection as well as other forms of noncontact testing.

The industry has also been leaning more heavily on boundary scan. But if you’re going to eliminate the standalone structural tester, where do you locate the boundary-scan controller? You still need it for some types of structural test as well as for flash and CPLD programming. So, why not add it to the functional tester? The idea has always been there, but now the implementation is moving forward.

Q: What other shifts in manufacturing test using PXI do you see?
A: One surprising area is that PXI is starting to move down from system- and module-level production test into component testing. Component-level test using PXI has been held back by a belief that dedicated ATE systems are needed for speed. But there has been increasing pressure to reduce test cost, and companies are looking for alternatives. This is opening the opportunity for PXI. You may need to give up some speed, but the test system will be significantly lower in capital cost, which could be a compelling tradeoff.

Q: What other opportunities for PXI?
A: Portable test using compact, ruggedized platforms is on the rise. Such systems are valuable to flight lines, repair depots, motor pools, and other field environments. There are so many functions available in PXI today that it can be a compelling alternative to stand-alone instruments if the PXI system is hardy enough.
Whose Fault is it?

That’s why test engineers have turned to Pickering Interfaces for years to solve their “Hardware In The Loop” fault simulation requirements to test and verify safety critical controllers. We provide a range of 12 different – with more to come – PXI-based fault insertion/simulation switching modules, including:

- First to market with a comprehensive choice of 2A, 5A, 10A, 20A and 30A scalable FI solutions
- Up to 64 channels of fault insertion in a single slot module
- BRIC FIBO matrices supporting up to 248 x 8 crosspoints for maximum fault insertion capability
- Extensive software support including Real Time: LabVIEW RT, QNX, Linux RealTime

Wherever safety critical simulation and testing of control modules is required, turn to Pickering for your next fault insertion test program.

See these products at [www.pickeringtest.com/FIBO/](http://www.pickeringtest.com/FIBO/)

www.pickeringtest.com

Pickering Interfaces Inc., Grants Pass, OR. Tel: +1 541 471 0700
E-mail: ussales@pickeringtest.com
(East Coast Regional Office) Woburn, MA. Tel: +1 781 897 1710

Direct Sales Offices in USA, UK, Sweden, Germany, Czech Republic, France and China
Customizing PXI test systems with FPGAs

By Sebastien Maury, Sundance Multiprocessor Technology

For the past decade, PXI-based systems have been successfully deployed for embedded signal-processing applications in data acquisition, industrial control, avionics, automated vision, medical instrumentation, and automated test. The high performance, modularity, and scalability of the PXI architecture have made it a compelling option for designers who require a rugged industrial form factor and real-time capabilities.

Today, designers and system engineers are increasingly deploying FPGA-based architectures to help deliver systems that are flexible and reconfigurable, support parallel processing, and offer a high data bandwidth. The integration of FPGAs into embedded signal-processing applications offers many benefits and advantages, but managing the diversity of I/O signals associated with FPGAs can make it difficult to interface devices to the external world.

Whether the I/O interfaces are digital, analog, single-ended, or differential, the preferred engineering solution is to allow designers to customize their FPGA-based hardware with the required I/O interfaces. Doing this while minimizing cost and customization is key.

New-generation FPGA products offer “plug-in” I/O hardware modules that are flexible enough to offer a wide range of interchangeable I/O functionality. These modules can directly interface to an FPGA, or other device, with reconfigurable I/O capability. They are configurable using programmable logic. Using FPGA mezzanine I/O modules can simplify system design, engineering time, and integration effort. Additionally, these modules can streamline the maintenance of the end product and increase the reusability of the main embedded signal-processing hardware units.

By combining FPGA cards and FPGA mezzanine I/O modules, users can design and deploy custom instrumentation. And with multichannel ADCs/DACs, RF front-ends, LVDS, LVTTL, Gigabit Ethernet, and serial interfaces, it is possible for test engineers to architect application-specific high-speed digital oscilloscopes, analyzers, arbitrary waveform generators, RF instrumentation, and vision systems. The combination of PXI and FPGA technology offers test engineers new options for building a modular, scalable system that meets their specific test needs.

Sebastien Maury is the Americas regional director at Sundance Multiprocessor Technology. sebastien.m@sundance.com.

Chroma unveils pin-electronics module

The 36010 pin-electronics module from Chroma ATE is a 100-MHz programmable module designed for characterizing and testing digital and mixed-signal ICs and electronics. The 36010 also supports scan pattern functions for scan test. Each module consists of a sequence pattern generator and a logic pin-electronics card with eight channels.

The sequence pattern generator, which provides more than 17 sequence commands to control the flow of pattern execution, is equipped with a 32-Mbyte sequence command memory. Each sequence pattern generator can support up to eight logic pin-electronics cards, allowing it to support up to 64 I/O channels and perform testing on eight devices simultaneously.

The per-pin timing generator in each logic pin-electronics card provides 32 sets of clock containing six programmable edges. In the analog function, the logic pin-electronics card has a tri-level driver and comparator with a 610-μV programmable resolution. It also offers active-load and high-voltage driver functions.


NI debuts controllers and a chassis

During NIWeek 2009 (August 4–6, Austin, TX), National Instruments introduced low-cost PXI Express chassis and controller options for automated test-and-measurement applications. Included in the new offerings are the $1499 NI PXIe-1073 chassis and the NI PXIe-8102/01 embedded controllers, which start at $2999 each.

The NI PXIe-1073 chassis, which features an integrated remote controller, features five PXI Express hybrid slots that accept both PXI and PXI Express modules and an integrated MXI Express controller with a PCI Express host controller card and cable. Built-in timing and synchronization connections are integrated into the backplane of the chassis.

The NI PXIe-8102/01 embedded controllers can address the needs of test engineers who require a PXI Express system that couples the PC and chassis in a self-contained system.

The NI PXIe-8102 features a dual-core 1.9-GHz Intel Celeron T3100 processor, and the NI PXIe-8101 includes a single-core 2.0-GHz Intel Celeron 575 processor.

GBATS: Geotest’s Basic Automated Test System

- Outstanding value and flexibility for functional test applications

- Standard application configurations for:
  - Mixed signal test
  - Digital test
  - Commercial and military avionics test

- The fully customizable PXI tester, for less green than you might imagine

Please visit us at AUTOTESTCON, Booth # 811
Using a real-time OS with PXI

By Richard A. Quinnell, Contributing Technical Editor

Given PXI’s roots in the PC field, it is no wonder that Windows is the dominant operating system for PXI system software. For some applications, however, Windows is not a good match to system requirements, and developers must employ another OS. Development teams seeking to move beyond Windows face challenges both in software availability and system programming, but new developments may offer a way past such problems. In fact, an evolving virtualization technology may permit a single test system to run more than one OS.

The drawbacks of Windows

One of the strengths of PXI is that the architecture is able to fully leverage technology advances coming from the fast-moving PC field. New processors, advanced interfaces, and development tools that arise in support of PC systems can quickly be incorporated into PXI modules and systems. The same is true of advances in system software such as Windows.

But Windows is a double-edged sword when it comes to system control. It enjoys wide support in terms of tools, applications, and developer expertise, but it also has key drawbacks. Two of the most critical for equipment developers are reliability and determinism.

Windows can have unpredictable timing and sometimes will crash for no readily apparent reason, a failing that is not tolerable in critical applications. “There is nothing mission critical about most manufacturing test systems,” said Wyatt Meek, director of business development at VI Engineering, “so they can run Windows, and if it crashes, they can simply reboot. But where operation is mission critical or there is critical control timing, you’ll want something else.”

As an example, Meek pointed to the JRETS (Jet and Rocket Engine Test System) that VI Engineering developed for Wyle Laboratories using multiple PXI systems to handle data acquisition and control. The system was designed to facilitate hot-fire testing of engines with as much as 50,000 lb of thrust, making consistent operation and well-controlled timing essential.

VI Engineering used the Phar Lap ETS RTOS (real-time operating system) instead of Windows, and used National Instruments’ LabView with the LabView Real-Time module as the programming environment. Reliability was a key reason for this choice.

“You won’t have a system crash with an RTOS if you implement it correctly,” said Meek. “The fear with Windows is getting the ‘blue screen of death’ in the middle of a test.” Meek also pointed to the maturity of LabView Real-Time, now at version 7, as a factor ensuring stable system operation.

In addition to reliability, the JRETS needed deterministic timing in its control paths. The variation in timing, or OS jitter, that Windows exhibits is typically 500 ms, and even when Windows is optimized for timing, the jitter can be more than 10 ms, according to Meek. An RTOS achieves jitter in the 1-to-10-ms range, making the system quicker to respond to errors, resulting in increased safety.

Developers want reliability

Even for systems that do not have such mission-critical requirements, however, some developers are looking for an alternative to Windows, according to Matthew Friedman, senior PXI platform manager at National Instruments. “A lot of users simply want more confidence in the reliability of their systems,” said Friedman. “They may also be looking for a higher degree of synchronization between the controls driving the test and the measurement.” Other reasons for choosing an alternative OS include vendor independence, version stability, and freedom from licensing fees.

If a PXI test system uses an external computer as the system controller, that computer can be running Windows, MacOS, Linux, or nearly any other OS that can send the appropriate commands. If the controller is embedded, however, off-the-shelf alternatives narrow.

Only a handful of manufacturers of PXI controllers support alternatives to Windows. Keithley Instruments, for instance, offers Linux for its controllers. NI has controllers with an embedded RTOS. MEN Mikro Electronik supports Linux, QNX, or VxWorks, depending on the controller model. Adlink has a Linux API (application programming interface) and driver library for its controllers and data-acquisition cards, and Team Solutions offers Linux drivers for the GPIB and PXI trigger functions built into its modular-CPU controller. Because the controller utilizes industry-standard plug-in CPU modules, however, OS support must come from the CPU module vendor.
If the controller is running an alternative OS, other system modules will require drivers appropriate to that OS. This can present developers with a challenge. “Not a lot of vendors have non-Windows drivers,” said Meek, “so there is a smaller range of resources available for developers.” Even when drivers are available, Meek noted, they may not support some of the modules that the Windows drivers do.

Embracing a Windows alternative can also limit your options for application software. “Developers should look at which programs support their OS,” said Meek, and he explained that if the test-control software does not run under the desired RTOS, developers will need to create their own test-sequencing engines.

There has been some industry activity to fill the gaps, at least for specific system configurations. LabView Real-Time, for instance, supports VISA (virtual instrumentation software architecture) drivers. So, if a module’s driver is VISA-compliant, it will work under LabView Real-Time.

The German company Sekas is offering software that makes the Rohde & Schwarz CompactTSVP (test system versatile platform), which is used in automotive and telecommunications test, compatible with Linux. The Sekas software—TSVP-LXLib—replicates the software infrastructure that IVI (Interchangeable Virtual Instrument) drivers need, making it easier for developers to port the drivers to Linux.

**Evaluate system needs**

For the most part, however, PXI developers seeking an alternative to Windows must evaluate their choice carefully. “Survey your current needs to ensure that you have support for the new OS,” said Gerardo Garcia, group manager for real-time software at NI. “Also, check to see if you are using OS-specific features such as ActiveX, which is only available under Windows. You need to make a full audit of what you are actually doing in your system.”

Developers who are seeking to use an RTOS should also think twice about simply making the change on their own. “Bringing up a controller board under a new OS can be painful,” said Garcia. “So, having out-of-the-box support for an RTOS from the board vendor is important.”

VI Engineering’s Meek also pointed out that moving a traditional PXI test application to a real-time system may require a learning curve. “You can’t just take normal LabView code and have it work in real-time,” said Meek. “You need to architect your
program to allow independent threads, set priorities, and the like. This might not be difficult for a specific test, but it gets tricky if you are trying to design a generic system with looping and such. This adds cost and complexity to the development effort. You have to ask if the advantages of an RTOS are worth it.”

Virtualization is on the horizon

Developers may not be facing an “either—or” OS choice in PXI system design for long, however. NI’s Friedman said the industry is on the verge of supporting the best of both worlds by embracing technology from the IT field. “Virtualization is an abstraction of hardware resources that allows multiple operating systems to run concurrently on a processor,” said Friedman. “It employs a hypervisor software layer underneath the OSes that keeps them separate.”

Virtualization technology takes advantage of the fact that all digital computing engines (processors) are Turing machines, which means that any processor can be programmed to mimic the behavior of any another processor regardless of structural and machine code differences. In virtualization, the hypervisor, also called the virtual machine monitor, runs at the processor’s foundation level to mimic multiple copies of system resources to higher-level software, creating VMs (virtual machines) that can each execute an OS and application code.

A hypervisor can provide a high degree of separation between VMs. An OS on one VM can crash, for instance, without affecting the operation of the others. The hypervisor can also coordinate access to system hardware resources such as memory and I/O so that each VM can function as though it has dedicated resources even if the resources are actually shared. Hypervisors can thus “split” a single processor into several functionally independent ones.

Many of the latest generation of Intel and AMD processors now have hardware features that help them efficiently run such hypervisors, and more such features are added with each generation, according to Friedman. Multicore processors, which are becoming the standard approach for attaining the highest CPU performance, are also good candidates for virtualization techniques. Thus, the technology is on the edge of being available for PXI.

The advantage of using virtualization in a PXI system is that it gives developers the ability to segment a controller’s functionality into multiple, independent parts. “You can keep the connectivity of Windows with one VM and use an RTOS for deterministic performance in another VM,” said Friedman. “This allows you to keep what you have under one OS but add more [functions] under another. It will allow for very innovative test system design.”

At the very least, virtualization can help developers seeking to adopt Windows alternatives. By keeping only the most critical functions under RTOS control and the rest under Windows, developers reduce their need for alternative drivers and other support. Such an arrangement also restricts the need for new software development for the real-time portions of the system. Developers thus may not need to move entirely beyond Windows to achieve their goals. They may simply be able to stretch their system’s reach a little.