Editor's Note: Three years after this article ran in the 2010 Automated Test Outlook, NI introduced the Vector Signal Transceiver, a PXI module that revolutionized RF instrumentation and created a new class of software-designed instruments that users can reprogram. At first, others in the industry called it "cute" and dismissed the notion that users would want to own the functionality of their instruments at that level. But the VST became the most successful hardware product from NI to date and redefined the future of instrumentation. If your organization isn't considering software-designed instrumentation yet, I strongly recommend it.
—Dr. James Truchard

Reconfigurable Instrumentation

Software-defined instrumentation, also known as virtual instrumentation, is based on a modular architecture that enables a high degree of reconfigurability. Software-defined instruments consist of modular acquisition/generation hardware whose functionality is characterized through user-defined software running on a host multicore processor. This basic model is ideal for most automated test applications in use today, but new technologies and test methodologies on the horizon are creating the need to push the reconfigurability down to the hardware to achieve required performance. One example of this is testing a modern RF receiver, where coding/decoding, modulation/demodulation, packing/unpacking, and other data-intensive tasks may need to occur inside a clock cycle of the device under test (DUT). In these cases, the software defined architecture needs to be flexible enough to incorporate user-programmable hardware—often a field-programmable gate array (FPGA)—to place the necessary intelligence inside the instrument. User-programmable instruments create an architecture where data can be acted upon in real time on the FPGA and/or processed centrally by the host processor (see figure). FPGAs are a key enabling technology because they combine the best parts of ASICs and processor-based systems. At the highest level, FPGAs are reprogrammable silicon chips. Using prebuilt logic blocks and programmable routing resources, engineers can configure these chips to implement custom hardware functionality. They can develop digital computing tasks in software and compile them down to a configuration file or bit stream that programs the FPGA components. In addition, FPGAs are completely reconfigurable and instantly take on a new personality when recompiled with a different configuration of circuitry.

> "The ability to customize the measurement hardware itself represents yet another milestone in the path toward a completely software-defined test system. In 10 years, we will wonder how we ever programmed test systems effectively without this capability."
>
> —Mike Santori, Business and Technology Fellow, National Instruments

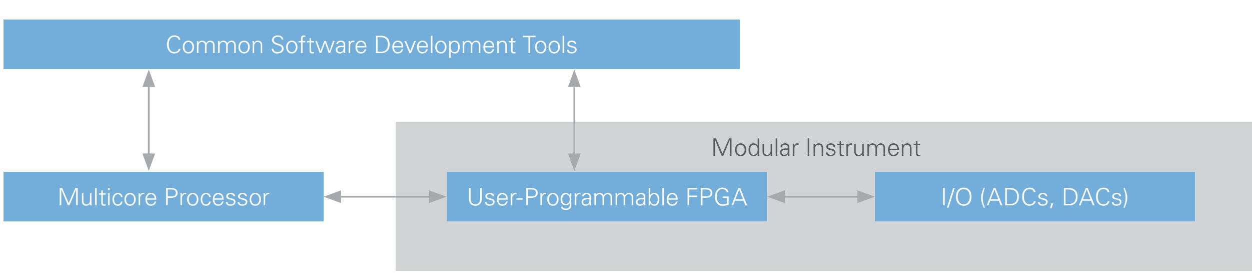
Beyond being user-programmable, FPGAs offer hardware-timed execution speed as well as high determinism and reliability. They are truly parallel so different processing operations do not have to compete for the same resources. Each independent processing task has its own dedicated section of the chip, and each task can function autonomously without any influence from other logic blocks. As a result, adding more processing does not affect the performance of another part of the application.

While FPGAs have been used inside instruments for over a decade, test engineers were seldom given access to embed their own algorithms on them. To be useful in a software-defined instrumentation context, FPGAs must be reprogrammable by the engineer in software; in other words, they should be used to push software programmability down into the hardware itself. In the past, FPGA technology was available only to engineers with a deep understanding of digital hardware design software, such as hardware description languages like Verilog or VHDL, which use low-level syntax to describe hardware behavior. Most test engineers do not have expertise in these tools. However, the rise of high-level design tools is changing the rules of FPGA programming, with new technologies that convert graphical block diagrams or even C code into digital hardware circuitry. These system-level tools that abstract the details of FPGA programming can bridge this gap.

Clearly, there are advantages to performing different types of processing on a host processor versus an FPGA. For example, an FPGA is generally well-suited for inline analysis such as simple decimations on point-to-point I/O, whereas complex modulation might achieve better performance running on a multicore processor due to the large amount of floating-point calculations required. The ideal solution for developing a software-defined test system is a single graphical system design development environment that provides the ability to quickly partition the processing on the host or an FPGA to see which offers superior performance.

This new software-defined architecture can meet application challenges that are impossible to solve with traditional methods such as the previous example that requires real-time decision making by the host to properly test the device. Instead, engineers can fully deploy the intelligence to the FPGA embedded on the instrument for pass/fail guidance. This is often the only way to supply the intense timing and determinism required by the DUT. Examples of this type of device include RFID tags, memory, microcontrollers, and engine control units (ECUs). For some applications, engineers also perform the communication over a protocol—wireless or wired—which requires a significant layer of coding and decoding before making a decision.

Reconfigurable instruments will continue to find more mainstream applications as test engineers continue to look for creative ways to reduce test time and system cost. Take, for example, a digitizer that has an FPGA inline with an analog-to-digital converter. An engineer can deploy functions to the FPGA such as filtering, peak detection, fast Fourier transforms (FFTs), or custom triggering. Not all data is created equal, but an FPGA-based digitizer can make quick decisions on which data is worthless and can be discarded and which data has value. This can ultimately reduce measurement time substantially. Test engineers in the military and aerospace industry have been early adopters of FPGA-based instrumentation through their synthetic instrumentation initiatives, but this technology also has potential for telecommunications, automotive, medical device, and consumer electronics applications.



Reconfigurable instruments provide a Host + FPGA configuration that delivers both performance and flexibility.