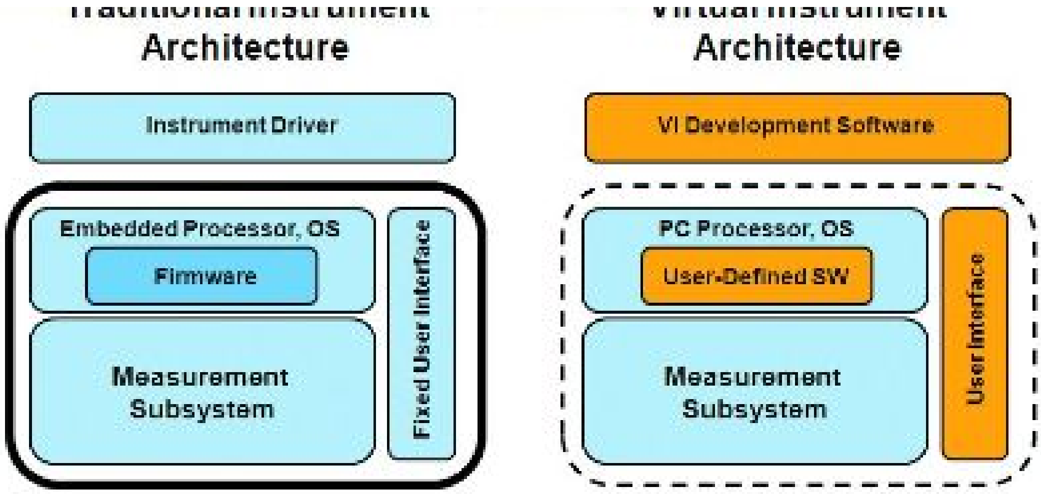
Design Talk: The New Face of Test Part 2

Automated Test Goes Software-Defined

by Richard McDonell, Senior Group Manager, National Instruments

Software-defined instrumentation is the new face of automated test. Scientists and engineers performing leading-edge research and designing custom measurement and control systems have used software-defined instruments, also known as virtual instruments, for more than 20 years. Software-defined instruments were critical for these often one-of-a-kind applications due to their unique system requirements. For example, engineers developing the first Mars Rover could not look up a part number in a catalog and order a spacecraft test and monitoring system. Instead, they relied on the flexibility of software-defined instrumentation to develop the specific data acquisition and control algorithms needed to design and monitor such a system[1].

The needs of today’s engineers are becoming increasingly similar to those of the Mars Rover engineers when it comes to testing the latest electronic devices. While spacecraft are not as ubiquitous as iPhones and Wii game consoles, software-centric next-generation electronic devices often require the flexibility of spacecraft test systems due to the thousands of unique design and test requirements they enable. Today’s electronic devices are complex computing machines with powerful embedded processors at their core. Hence, software-defined devices are becoming the new standard, and for good reason. With software-defined devices, engineers can improve design reuse by developing entire product lines from the same hardware design, enable faster time to market through fewer board spins, and resolve design flaws by downloading the latest software updates at the click of a button. However, keeping up with the light-speed design cycles and ever-changing test requirements is the new

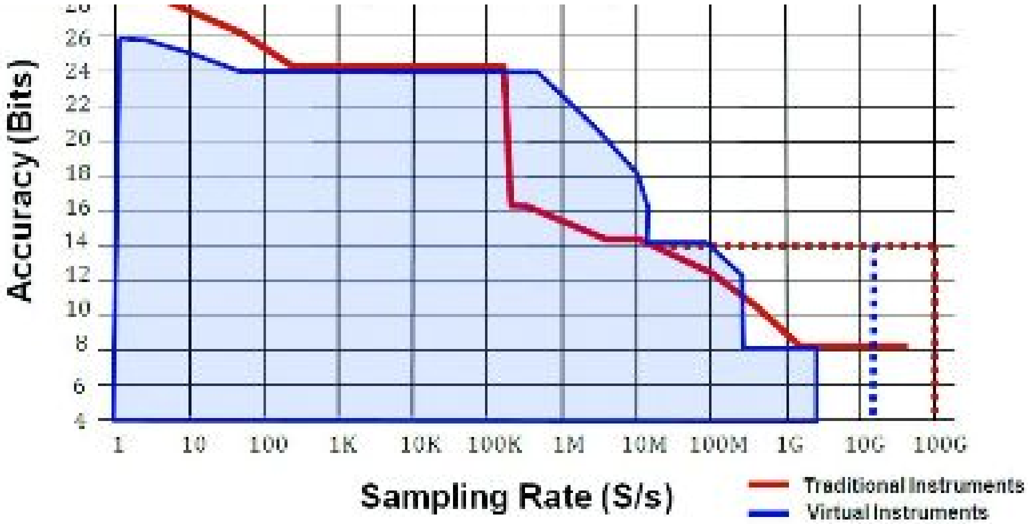


gravity-defying challenge that today’s design and test engineers face.

Software-defined Instrumentation

The growth in software-defined instrumentation reflects the need for increased flexibility and a higher degree of measurement automation throughout electronic design and production. Engineers can no longer afford delays in their product cycles while traditional instrument vendors design a new box to test the latest wireless standard, or be limited to the fixed, hardware-defined personality of a box instrument. Today’s design and test engineers need the flexibility to perform all of their tests using a common instrumentation core and the ability to apply their own algorithms as required by their software-defined devices under test.

The industry-standard software and hardware for developing software-defined instruments are NI LabVIEW and the open, multivendor PXI hardware standard[2]. Currently, more than 70 global vendors offer more than 1,500 PXI devices that can be used with LabVIEW, C, or C++ to develop software-defined instrumentation. LabVIEW, for example, provides many built-in measurement algorithms and data I/O libraries to enable



software-defined instruments to perform the same off-the-shelf DC to RF measurements as a traditional box instrument (<--- Figure 2). The advantage of software-defined instrumentation, however, is the ability to combine standard measurements with custom data processing and measurements using the same modular hardware. For example, an NI PXI-4071 digital multimeter (DMM) can function as a 7.5 digit DMM as well as a 1.8 MS/s, 1000 V digitizer for custom analysis of DC-coupled waveforms. Likewise, an NI PXIe-5663 6.6 GHz vector signal analyzer can be used to test dozens of wireless standards including WLAN, WiMAX, DVB-T, and GPS as opposed to the traditional approach of using a dedicated stand-alone instrument for each standard (Figure 3 --->).

Fast, Flexible Automation Is Key

Software-defined instruments are inherently flexible and easily automated. This enables today’s product design teams to streamline their development process by reducing the number of hours consumed in manual testing and by minimizing the amount of instrumentation required in the lab and on the production floor. Software-defined instrumentation is successfully used across all industry segments today, including consumer electronics, communications, aerospace and defense, and automotive. All of the top electronics companies and contract electronics manufacturers use software-defined instrumentation in many of their complex and high-volume design and production areas. Furthermore, an estimated 20,000 PXI-based systems containing more than 100,000 software-defined instruments will be deployed in 2008, and market analyst firm Frost & Sullivan expects the PXI market to continue growing at up to 23 percent CAGR through 2012.3 “The adoption of tools such as PXI is an indicator that companies recognize the benefits of moving toward software-defined instruments,” said Kiran Unni, Frost & Sullivan Measurement & Instrumentation research manager. “The savings being realized in capital equipment, system development, and improvements in system efficiency all contribute to reducing the per-unit cost of test, directly influencing the bottom line.”[4]

New Technology Fuels Growth

Software-defined instrumentation uses the latest technology advancements to deliver new levels of measurement performance and to identify new areas of productivity where traditional instrumentation solutions are unable to keep up with the fast moving needs of test. Three new technology and application areas fueling rapid industry growth in software-defined instrumentation today are wireless test, FPGA-based reconfigurable I/O, and multicore processing.

Wireless Test

RF and wireless test is among the fastest growing electronic areas and is one of the most challenging for design and test engineers. In addition to learning how to design and test wireless devices, engineers often find three or more wireless standards in today's devices. Soon, RF instrumentation could become as ubiquitous as general-purpose instruments. This presents a major challenge for engineers who cannot afford bottlenecks in their design cycles due to lagging availability of instruments for testing the latest RF standards, and for those who cannot absorb the increased cost of stand-alone RF boxes for testing each wireless standard. Fortunately, software-defined instrumentation offers a tremendous benefit to engineers testing wireless standards. With software-defined instrumentation, engineers can test multiple standards using a common modular instrumentation core and implement custom wireless protocols and algorithms in their test systems regardless of the maturity of a new wireless standard. Automated measurement speed is also increased by up to 20X with multicore processing and high-speed bus interfaces used in the RF modular instruments.

FPGA Reconfigurable I/O

Another area experiencing rapid expansion in software-defined instrumentation is the increase in system-level tools for field-programmable gate arrays (FPGAs). Leading vendors are including FPGAs on modular instruments and giving engineers the access in software to reprogram their virtual instruments according to their requirements. For example, test engineers can use LabVIEW to embed a custom algorithm into the modular instrument to perform in-line processing inside the FPGA or emulate part of the system that requires a real-time response. As electronic devices become more complex, testing will become more integrated in the design process and user-defined measurements inside software-defined instrumentation will become even more important.

Multicore Processing

Today's electronic devices process unprecedented amounts of data. The multifunction capabilities of these smart devices further compound the data rates in the devices and the number of tests that must be performed to validate the design and final manufactured products. Software-defined instrumentation uses the latest multicore processors and high-speed bus technologies, such as PCI Express, to ensure engineers are able to generate, capture, analyze, and process the gigabytes of raw data required to properly design and test today's electronics products.

Conclusion

Software-centric devices have streamlined the design process for today's electronic devices. Software-defined instruments address the required flexibility, performance, and automation to quickly and efficiently test such devices. Engineers are encouraged to learn more about software-defined instrumentation from National Instruments by visiting www.ni.com [1].

References

1 National Instruments LabVIEW Software Monitors Health of Mars Pathfinder Sojourner Rover, 1997,

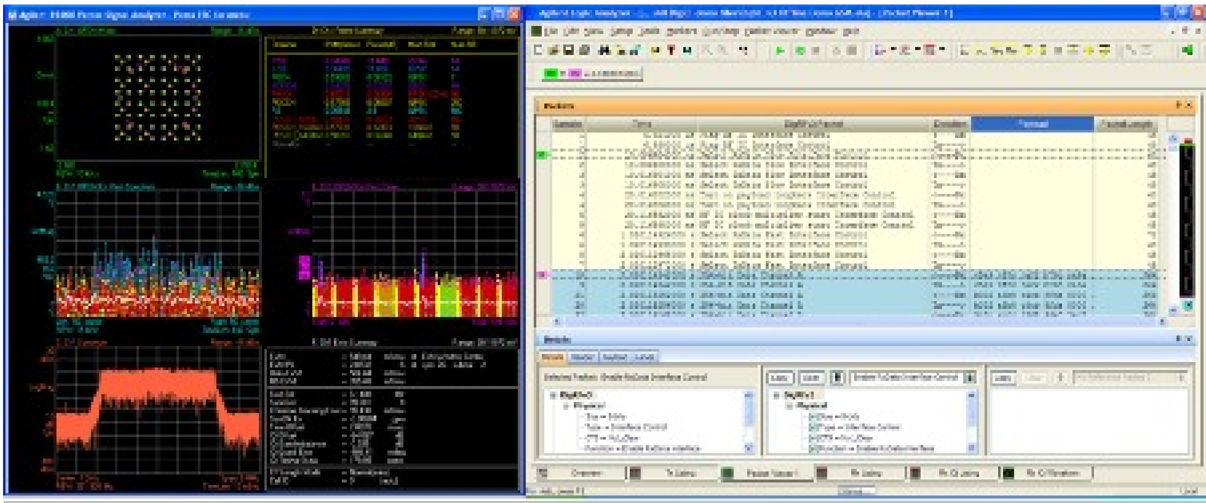
http://findarticles.com/p/articles/mi_m0EIN/is_1997_July_18/ai_19593795 [2]
2 PXI Systems Alliance, http://pxisa.org [3]
3 PXI Systems Alliance Announces 10th Anniversary of PXI, 2007, http://findarticles.com/p/articles/mi_pwwi/is_/ai_n19525793 [4]
4 Test and Measurement Industry Trends Toward Software-Defined Instrumentation and Use of Multicore-Enabled Test Systems, 2008, http://digital.ni.com/worldwide/bwcontent.nsf/web/all/96ACA34134A29F05862573C500539AD7 [5]

Test Challenges When Integrating High Speed Serial Links In RF-IC, Part 2

by Jean-Manuel Dassonville, Agilent Technologies, Inc.

Part one of this article — which appeared in the November 1 Design Talk section of ECN — reviewed technical design and test challenges associated with the integration of high speed serial links in mobile RF-IC. As soon as the DigRF link is operating properly, the next phase of the test process consists of moving up on the protocol stack and then starting the tests in the wireless domain. Part two of this article continues to describe how to set up a combination of DigRF protocol analyzer/exerciser with spectrum analyzer and sources to address these challenges. The complete article can be viewed online at www.ecnmag.com/design-talk-november-2008.aspx [6].

Migrating from Analog to Digital Test
If the link between the BB-IC and RF-IC is an analog IQ link, the test environment for modulation measurements is based on a signal source and a signal analyzer, combined with vector signal generation and vector signal analysis software. As this link migrates to DigRF, the hardware elements of the test environment are replaced by DigRF exercisers and analyzers. The digitized IQ waveform is inserted or extracted from the payload of DigRF packets.



As designs may include a combination of digital and analog IQ links (legacy inputs), it is important for the test environment to show consistent measurements independently of the link being used. The optimal test architecture includes common vector signal generation and vector signal analysis software integrated into both analog and digital analysis and stimulus hardware.

Mixed Traffic Generation and Analysis
The traffic flowing on a DigRF link provides the homogeneous digital IQ waveform information and is used to transport a heterogeneous mix of various traffic patterns. These are the RF-IC initialization commands and control commands that are used to adjust some parameters of the RF-IC device during its operation, as well as synchronization of words and acknowledgment of messages.

From a test stimulus point of view, the challenge consists of defining each pattern and building realistic stimulus that aggregates the various patterns in a main traffic flow. Time determinism is crucial as some patterns must be sent at very precise time intervals. The proper test environment includes “signal insertion” software that works in conjunction with the vector signal generation software and automates the construction of a complete stimulus file from these different elements. If one parameter needs to be changed, the signal insertion software will ensure that all other parameters remain constant.

During the debug and system integration phase, these various traffic patterns influence the overall behavior of the system under test. If one pattern is not set up properly, the entire system may fail. The faulty pattern may be a bad signal, an improper digitization of IQ data, a wrong command, or synchronization patterns that are not sent on time. In order to get insight on the system activity, the DigRF link is the critical visibility point. A key element of the test environment is a DigRF analyzer that performs non intrusive real-time monitoring of the link, extracts digital IQ waveform from other traffic flows and provides cross-domain insight in both wireless and digital domains, as shown here.

Conclusion
The adoption of high speed standard interconnects in mobile handsets brings tremendous benefits in terms of performance and power consumption. It is driving major changes in the test methodology from chip turn-on to system integration.

True cross-domain test approaches are essential to debug and validate DigRF enabled mobile systems.

See next page for an article from Digitaltest... **[7]**

Links:
[1] http://www.ni.com/
[2] http://findarticles.com/p/articles/mi_m0EIN/is_1997_July_18/ai_19593795
[3] http://pxisa.org/
[4] http://findarticles.com/p/articles/mi_pwwi/is_/ai_n19525793
[5] http://digital.ni.com/worldwide/bwcontent.nsf/web/all/96ACA34134A29F05862573C500539AD7
[6] http://www.ecnmag.com/design-talk-november-2008.aspx
[7] http://www.ecnmag.com/Design-Talk-New-Face-of-Test-Part2-2.aspx