Designing Next Generation Test Systems
An In-Depth Developers Guide
Designing Next Generation Test Systems

An In-depth Developers Guide
**Contents**

Section 1  
Executive Summary

Chapter 1  
Designing Next Generation Test Systems  
Increasing Design Complexity ................................................................. 1-1  
Shorter Product Development Cycle ......................................................... 1-2  
Increasing Test Cost and Decreasing Test Budget ................................. 1-2  
Designing Next Generation Test Systems ............................................... 1-2  
Test System Management Software ......................................................... 1-4  
Conclusion: Designing Next Generation Test Systems .......................... 1-8

Section 2  
Guidelines for Designing Next Generation Test Systems

Chapter 2  
Developing a Modular Software Architecture for Reducing Development  
Cycles and Cost  
Industry Trends and Challenges ............................................................... 2-1  
Defining a Modular Test Software Framework ......................................... 2-1  
The National Instruments Modular Test Software Framework ............... 2-3  
Application Development Environments (ADE)...................................... 2-5  
Measurement and Control Services ......................................................... 2-6  
Conclusion: Developing a Modular Software Architecture for  
Reducing Development Cycles and Cost ................................................ 2-8

Chapter 3  
Choosing the Right Software Application Development Environment  
Factors to Consider When Selecting an ADE ........................................... 3-1  
LabVIEW Graphical Programming Environment .................................... 3-3  
LabWindows/CVI, an ANSI C Development Environment .................... 3-7  

Chapter 4  
Hybrid Systems – Integrating Your Multivendor, Multiplatform Test Equipment  
The Challenge for Test Systems Developers ............................................ 4-1  
What Is a Hybrid System? ....................................................................... 4-1  
Making the Right Software Decisions ..................................................... 4-3  
Designing Your System ......................................................................... 4-5  
Conclusion: Building Hybrid Systems .................................................... 4-6
### Chapter 9
**Maximizing Accuracy in Automated Test Systems**
- Understand Instrument Specifications ................................................................. 9-1
- Consider Calibration Requirements ................................................................. 9-2
- Be Aware of the Operating Environment ...................................................... 9-3
- Use Proper Fixturing ....................................................................................... 9-5
- Take Advantage of Synchronization .................................................................. 9-7
- Conclusion: Maximizing Accuracy in Automated Test Systems ......................... 9-8

### Chapter 10
**Designing and Maintaining a Test System for Longevity**
- Software Considerations .................................................................................. 10-1
- Hardware Considerations .................................................................................. 10-3
- Maintenance and Support Considerations ...................................................... 10-4
- NI Solution for Building and Maintaining Test Systems for Longevity .............. 10-4
- Conclusion: Designing and Maintaining a Test System for Longevity ............... 10-11

### Section 4
**Case Studies and Customer Applications**

#### Chapter 11
**Software-Defined Radio Architecture for Communications Test**
- Flexible Software-Defined Communications Test ........................................... 11-3
- PXI – An Ideal Platform for Software-Defined Communications Test ............. 11-6
- Software-Defined Communications Systems Provide a Future-Proof Platform .. 11-7

#### Chapter 12
**Microsoft Uses NI LabVIEW and PXI Modular Instruments to Develop Production Test System for Xbox 360 Controllers**
- Designing Powerful Controllers for a New Generation of Gaming .................. 12-1
- PXI Modular Instruments for Design Validation and Production Test ................ 12-2
- NI LabVIEW Interfacing with Microsoft SQL Server, TCP/IP, and ActiveX Controls ................................................................. 12-3
- Microsoft Sees Results Using NI LabVIEW and PXI Modular Instruments ...... 12-3

#### Chapter 13
**U.S. Air Force Increases Mission-Capable Rates with PXI**
- U.S. Air Force Increases Mission-Capable Rates with PXI ............................. 13-1
- Using PC-Based Software and Hardware to Lower Costs ............................... 13-2
- Reducing Test System Size by 50 Percent .................................................. 13-2
Chapter 14
Sanmina-SCI Exceeds Throughput Goals with PXI Tester and Multithreaded Software

Test System Requirements ................................................................. 14-2
Compact, High-Speed Test Solution .................................................. 14-2
Exceeded Yield Expectations ............................................................ 14-3
References ......................................................................................... 1-1
Executive Summary
Welcome to the Designing Next Generation Test Systems Developers Guide. This guide is the first in a series of whitepapers designed to help you develop test systems that lower your cost, increase your test throughput, and can scale with future requirements.

Test managers and engineers use automated test systems in applications ranging from design validation to end-of-line production test to equipment repair diagnostics with the goal of ensuring the quality and reliability of a product that reaches the end customer. They can use automated test systems to perform simple "pass" or "fail" tests or a whole range of product characterization measurements. As it grows exponentially more expensive to detect flaws later in the design cycle, automated test systems have quickly become an even more important part of the product development process. This first document titled "Designing Next Generation Test Systems" describes the challenges that continue to pressure engineering teams to reduce the cost and time of test. It also provides insight into how test managers and engineers are overcoming these challenges by building modular, software-defined test systems that significantly increase test system throughput and flexibility while reducing overall cost.

Today's test engineers face a new set of pressures. Test engineers are presented with the following realities of today's product development environment:

- Product designs are more complex than their previous generations
- Development cycles are shorter to stay competitive and meet customer demand
- Budgets are decreasing while product testing is becoming more expensive

**Increasing Design Complexity**

One of the most visible trends is the increased device complexity. For example, the consumer electronics, communications, and semiconductor industries continue to drive the convergence of digital imaging/video, high-fidelity audio, wireless communication, and Internet connectivity into a single product. Even the automobile has integrated sophisticated car entertainment and information systems, safety and early warning systems, and body and engine control electronics. Test system designs must be flexible enough to support the wide variety of tests that differ between product models but also scalable to accommodate a larger number of test points as new measurement functionality is required.
Shorter Product Development Cycle

The ever-increasing demand for the latest product or technology combined with the competitive nature of being the first to market puts pressure on design and test engineering teams to shorten their product development cycles. To be successful, engineering teams need to develop new test strategies to decrease test time and improve efficiencies from design through production.

Increasing Test Cost and Decreasing Test Budget

Increase device functionality often leads to a more expensive and time-consuming test process. However, the cost to build each function is decreasing, which challenges test engineering departments to reduce its cost and budget, as shown in Figure 1-1. As a result, engineers must develop test strategies that reduce cost by increasing the throughput of their test systems, reducing the maintenance and upgrade costs, and lowering the required capital investment.

![Cost of Silicon Manufacturing and Test](image)

**Figure 1-1.** Data from SIA illustrates that the cost of silicon (or device function) has decreased over time, but the cost to test continues to increase.

Designing Next Generation Test Systems

To meet the challenges they face with increased device complexity, shorter development cycles, and decreased budgets, test managers and engineers are being forced to abandon traditional test design strategies based on traditional box instruments or "big iron" propriety ATE systems. These stand-alone instruments lack the flexibility needed as the software processing and user interface are defined by the supplier and can only be updated by the supplier through firmware. This makes it difficult to perform measurements not defined in the instrument’s firmware and measurements for new standards or to modify the system if requirements change. These devices also lack necessary integration capabilities such as data streaming and synchronization because they are designed primarily to
work as stand-alone instruments and not for integrated system use. Proprietary ATE systems, such as highly integrated production chip testers, provide the performance needed but are typically cost-prohibitive and can leave engineering teams vulnerable to obsolescence and untimely system redesigns.

In response to these trends, test managers and engineers are now implementing modular, software-defined test architectures based on widely adopted industry standards to provide:

- Increased test system flexibility deployable to a variety of applications, business segments, and product generations
- Higher-performance architectures that significantly increase test system throughput and enable tight correlation and integration of instruments from multiple suppliers including precision DC, high-speed analog and digital, and RF signal generation and analysis
- Lower test system investments by reducing initial capital investment and maintenance cost while increasing equipment use across multiple test requirements
- Increased test system longevity based on widely adopted industry standards that enable technology upgrades to improve performance and meet future test requirements

National Instruments, a leader in automated test, is committed to providing the hardware and software products engineers need to create these next generation test systems.

This in-depth developers guide includes the information you need to design your next test system architecture. This introduction describes a test system architecture, figure 1-2, that provides engineers with a strategy to meet challenges related to increased device complexity, shorter development cycles, and decreased budgets.
Test System Management Software

An automated test system requires the implementation of several tasks and measurement functions – some specific to the device under test (DUT) and others repeated for every device tested. To minimize maintenance costs and ensure test system longevity, it is important to implement a test strategy that separates the DUT-level tasks from the system-level tasks so engineers can quickly reuse, maintain, and change test programs (or modules) created throughout the development cycle to meet specific test requirements.

In any test system, there are often operations that are different for each device tested and operations that are common for each device tested such as system-level tasks.

<table>
<thead>
<tr>
<th>Operations different for each device</th>
<th>Operations common for each device</th>
</tr>
</thead>
<tbody>
<tr>
<td>Instrument configuration</td>
<td>Operator interfaces</td>
</tr>
<tr>
<td>Measurements</td>
<td>User management</td>
</tr>
<tr>
<td>Data acquisition</td>
<td>DUT tracking</td>
</tr>
<tr>
<td>Results analysis</td>
<td>Test flow control</td>
</tr>
<tr>
<td>Calibration</td>
<td>Storing results</td>
</tr>
<tr>
<td>Test modules</td>
<td>Test reports</td>
</tr>
</tbody>
</table>

Figure 1-2. National Instruments offers a complete hardware and software solution for designing automated test systems.
Some companies have written their own test executives and spent valuable engineering resources to develop test management software from the ground up. This often results in reduced productivity and ties up resources over time for software maintenance. To maximize productivity, engineering teams should use commercially available test management software, such as NI TestStand, to reduce the development of operations that are common for each device. By using this software, engineers can focus their development efforts on the operations that are different for each device.

For more information, view the whitepaper titled, "Developing a Modular Software Architecture."

**Application Development Software**

The application development environment (ADE), such as National Instruments LabVIEW and LabWindows/CVI, plays a critical role in test system architectures. With these tools, test system developers can communicate to a variety of instrumentation, integrate measurements, display information, connect with other applications, and much more. Ideally, the ADE(s) used to develop test and measurement applications provide ease of use, compiled performance, integration of a diverse set of I/O, and programming flexibility to meet the requirements for a range of applications.

Ease of use goes beyond how quickly someone can get up and running. With easy-to-use ADEs, developers can easily integrate processing routines with multiple measurement devices, create sophisticated user interfaces, deploy and maintain applications, and modify the application as product designs evolve and system needs expand.

For more information, view the whitepaper titled, "Choosing the Right Software Application Development Environment."

**Measurement and Control Services**

Measurement and control services play an important role by providing connectivity to various hardware assets in the system, system configuration, and diagnostic tools. For example, NI Measurement & Automation Explorer (MAX) automatically detects hardware assets including data acquisition and signal conditioning hardware; GPIB, USB, and LAN-controlled instruments; PXI systems; VXI devices; and modular instrumentation so developers can configure them all in one place. Integrated diagnostic tests ensure that devices function properly, and test panels provide a quick way to check the functionality of the hardware before developers begin programming. Measurement and control services should also provide integration with the application development software layer through application programming interfaces (APIs) so developers can easily program their instruments. In fact, the components of this services software – hardware drivers, application programming interfaces (APIs), and a configuration manager – must seamlessly integrate within the ADEs to maximum performance, increase development productivity, and reduce overall maintenance.

For more information, view the whitepaper titled, "Developing a Modular Software Architecture."
Computing and Measurement Bus

At the center of every automated test system today is a computer in the form of a desktop PC, server workstation, laptop, or embedded computer as used with PXI and VXI. An important aspect of the computing platform used is the ability to connect (and communicate) to the wide variety of instruments in a test system. There are several different instrumentation buses available for stand-alone and modular instruments including GPIB, USB, LAN, PCI, and PCI Express. These buses have differing strengths making some more suitable for certain applications than others. For example, GPIB has the widest adoption for instrument control and wide availability of instrumentation; USB provides wide availability, easy connectivity, and high throughput; LAN is well-suited for distributed systems; and PCI Express provides the highest performance.

The widespread use of the PC has generated the proliferation of high-performance internal buses including PCI and PCI Express, which provide the lowest latency and highest data throughput or bandwidth. The PCI bus provides up to 132 MB/s of bus bandwidth and PCI Express, an evolution of PCI, can scale up to 4 GB/s to meet growing bandwidth needs while still providing complete software compatibility with PCI. Figure 1-3 illustrates the latency and bandwidth performance of the most popular instrument control buses.

![Figure 1-3. A comparison of various instrument control buses. PCI and PCI Express provide the best bandwidth and latency or overall throughput performance.](image)

For more information, view the whitepapers titled: "Hybrid Systems: Integrating Your Multi-Vendor, Multi-Platform Test Equipment" and "Instrument Bus Performance: Making Sense of Competing Bus Technologies for Instrument Control."
Measurement and Device I/O

Fundamentally, there are two types of instrumentation architectures today – traditional and virtual. Figure 1-4 illustrates the similarities in these two approaches. Both have measurement hardware, a chassis, a power supply, a bus, a processor, an OS, and a user interface.

The most obvious difference from a hardware standpoint is how the components are packaged. A traditional, or stand-alone, instrument puts all of the components in the same box for every discrete instrument. The measurement functionality, analysis, displays, and control of the instruments is defined by the supplier.

By contrast, modular, software-defined virtual instruments incorporate general-purpose measurement hardware that helps users go beyond the standard capabilities and define their own measurements and user interfaces in software. With a modular approach, engineers can define the test system measurement functionality and build systems that scale to meet future demands. Through a modular, software-defined approach, users make custom measurements, perform measurements for emerging standards, or modify the system if requirements change (for example, to add instruments, channels, or new measurements). This combination of flexible, user-defined software and scalable hardware components is the core of modular instrumentation.

For more information, view the whitepapers titled: "Understanding a Modular Instrumentation System for Automated Test" and "PXI: The Industry Standard Platform for Instrumentation."
Conclusion: Designing Next Generation Test Systems

Increased device complexity, shorter development cycles, and decreased budgets provide an opportunity for engineering teams to re-evaluate their current automated test strategies and look for areas to increase efficiency and reduce cost. When designing next generation test systems it is important to incorporate strategies that increase system flexibility, deliver higher measurement and throughput performance, lower test system cost, and expand longevity. Modular, software-defined automated test systems overcome the shortfalls of past solutions based on stand-alone instrumentation or cost-prohibitive propriety ATE systems. A modular hardware platform based on widely adopted industry standards platform, such as PXI, allow engineers to develop scalable test systems that tightly integrate the functionality from a variety of instrumentation suppliers. In addition, it also allows engineering teams to integrate current equipment investments lowering the initial cost of implementation. Along with software-defined measurements that make use the latest PC technology such as multiple core processors and PCI Express, next generation test systems can significantly improve throughput performance and scale to meet the demands of different product generations and business segments.

Several companies already implement a modular, software-defined test system strategy and prove the return on their investment. For example, Microsoft developed the test system for the Xbox 360 controllers based on NI LabVIEW and PXI Modular Instruments that resulted in a test system that operates twice as fast that their previous generation. The U.S. Air Force developing test architecture that supporting their premier fighter aircrafts. Using a PC-based software and hardware architecture, they were able to lower costs and reduce size of test systems by 50 percent. Sanmina-SCI builds FDA approved medical device test systems based on NI TestStand and PXI that exceeds their requirements to test over 83,000 devices per week and exceeds their yield production requirements by 95 percent.