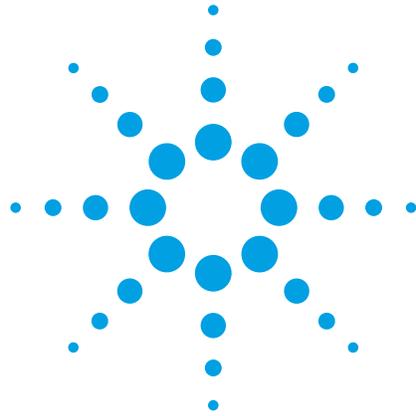


# Building Hybrid Test Systems, Part 2

## Ensuring success in two common scenarios



Application Note 1465-33

In building a hybrid test system, the best choices of hardware architectures depend on the overall situation, the specific test requirements and your priorities. When you migrate to a hybrid system that includes LXI along with GPIB, VXI, PXI, or any other architecture, two scenarios are most typical:

- **Minimum development time**  
You're probably combining old and new equipment and want to quickly assemble a working system so you can start testing the DUT as soon as possible. System assembly, programming and debugging are necessary, but you want to minimize the time spent on these activities. You want good performance, but it is not your top priority.

- **Maximum overall performance**  
You're performing high-volume testing and the per-unit test time will dominate the overall cost of test. You are willing to spend more time up front to get better system performance.

Hybrid systems that include multiple architectures are a viable approach to either scenario.<sup>1</sup> This application note describes these two scenarios and presents suggested steps that will help you minimize development time and maximize overall system performance.

---

<sup>1</sup> As described in the companion application note (5989-8175EN), using no more than two architectures is the best way to ensure success when building a hybrid test system.



## Example 1 Minimizing development time

We recommend this sample process as a pathway to success when you are combining older (e.g., GPIB) and new-generation (e.g., LXI) test equipment. For this example, the goal is to quickly create a working system so you can start testing your device as soon as possible. This process minimizes the time spent on system assembly, programming, debugging, and performance tuning.

Creating the hybrid system requires just five steps:

1. Replace an old instrument with a compatible new instrument.
2. Use a multiport instrument (e.g., one equipped with GPIB and LAN/LXI ports).
3. Move the instrument to LAN using the GPIB-compatibility mode built into most LXI instruments.
4. Move the remaining GPIB instruments to the LAN with a GPIB gateway or converter (Figure 1). Automatic capabilities will discover the converter and then find the attached instruments.
5. Do some basic analysis of performance, perhaps using sockets to tune.

That's the basic procedure. Let's look at each step in more detail.

### Step 1 Replace old with new

In many cases, the simplest migration step is to simply replace an existing instrument with a new model that is function- and command-compatible. This is beneficial when the replacement instrument provides new capabilities or significantly better performance (e.g., speed or resolution).

In many cases, this is a drop-in replacement. However, a few common issues must be addressed:

- **Instrument identity**  
If your system software specifically checks the instrument identity (ID) you will need to change it. Also, the

initialization routines of many IVI drivers have a flag to test the ID. This must be set to False because the instrument will return a different ID. Figure 2 shows how to turn off instrument ID checking for an IVI driver.

- **Numerical precision**  
If your system checks against precise test limits, it is best to verify that the new instrument will provide compatible results.
- **Instrument speed**  
Many newer instruments are significantly faster than those they replace

and this can cause problems in program timing. It can also be an opportunity to retune your program for more performance. (Note: Example 2 will address this point in more detail.)

As a plug-compatible example, you can replace the Agilent 34401A DMM with the newer 34410A DMM. In this case, you can activate the compatibility mode using the SCPI command `SYSTEM:LANGuage "34401A"`, thereby ensuring that your program will continue to work as expected. Many other instruments use similar commands to activate their compatibility modes.

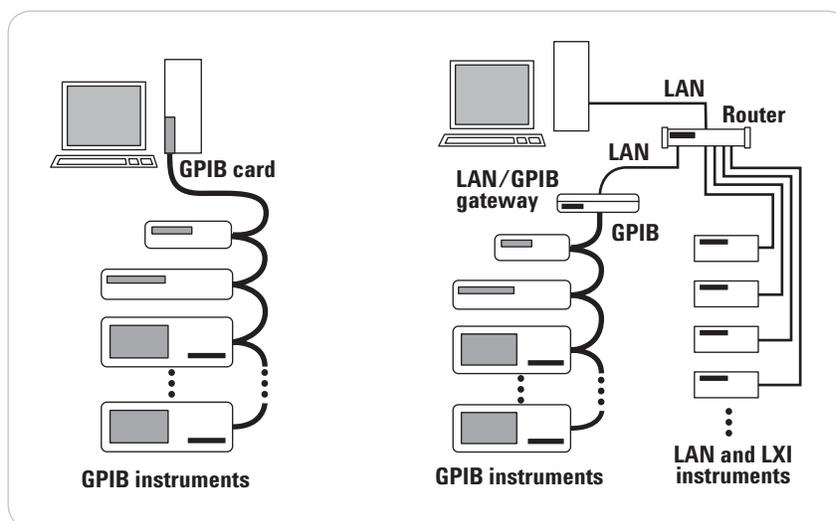


Figure 1. An all-GPIB system (left) can be easily converted to a hybrid structure (right).

#### Original IVI-COM declaration for 34401A:

```
Dim DMM As VisaComLib.FormattedIO488
Set DMM = New Agilent34401
DMM.Initialize "GBIB0::23::INSTR", True, True, ""
```

#### New IVI-COM declaration for 34401A when using VXI-11:

```
Dim DMM As VisaComLib.FormattedIO488
Set DMM = New Agilent34401
DMM.Initialize "TCPIP1::169.254.4.10::inst0::INSTR",
False, True, "QueryInstrStatus = True"
```

**Set instrument ID false**

Figure 2. When using compatibility mode with a replacement instrument, setting ID checking to "False" will ensure proper operation of your test program.

## Step 2

### Use a multiport instrument

Many LXI instruments are equipped with multiple input/output (I/O) ports—LAN, USB, GPIB (Figure 3). You can use this to your advantage to verify instrument programming and operation in a GPIB environment before moving to the LAN interface. First, connect to the instrument through GPIB and get it working. This lets you verify operation in a familiar environment. Then continue to Step 3.



Figure 3. Many of today’s instruments have multiple I/O ports: the Agilent 34980A switch/measure unit shown here is equipped with GPIB, LAN and USB ports.

## Step 3

### Move the instrument to LAN

Agilent LXI instruments equipped with multiple I/O ports provide a compatible programming mode between GPIB and LXI (called VXI-11). This allows you to simply replace the GPIB address declaration with a LAN declaration. For example, changing from GPIB address 9 to LAN address 156.140.71.39 requires that you replace the string “GPIB0::9::INSTR” with “TCPIP0::156.140.71.39::INSTR” in the declaration section of your test program. If you aren’t using a declaration, then a search-and-replace operation can perform the update throughout your program.

Note that the VXI-11 compatibility mode adds an additional layer of software so may not be appropriate for performance-critical operations. In Example 2, we’ll show how to reprogram for more speed.

In this step, you’ll also need to connect the instrument appropriately through the LAN. We recommend you set up a private network behind a LAN router. You can also put your computer on this network. By configuring the router for network address translation (NAT), the computer will have full access to your network while isolating your instruments from other network traffic.

## Step 4

### Move remaining GPIB instruments to the LAN

You’ve started the transition to LAN and now it’s time to move everything else. You can put your GPIB instruments “on the LAN” through a LAN/GPIB gateway. One proven alternative is the Agilent E5810A LAN/GPIB gateway, which enables the move without the need for programming changes. Instead, everything is taken care of by the Agilent IO libraries. Assuming your existing GPIB card is “GPIB0”, here are the steps:

- Remove your existing GPIB card from the PC (you can’t have two GPIB0 cards).
- Connect your PC to one port of your LAN router and connect the E5810A to another port. Connect your GPIB instruments to the GPIB port on the E5810A. See Agilent Application Note 1465-10, *Using LAN in Test Systems—Network Configuration*, for information on setting up your router.
- From your computer desktop or Start menu, start the IO Libraries (Figure 4).
- Click on the LAN port, then click **Add Interface**.
- In the popup dialog box, select **Remote GPIB (via E5810 or Remote IO Server)** and click **Add**.
- A second dialog box will appear: Set the ID as GPIB0, test the connection, and discover the connected instruments.

Be aware that there may be slower I/O performance when using the gateway. The reason: Any type of format conversion uses additional time and resources. However, unless a large number of time-critical operations pass through the converter, the difference probably won’t be noticeable. If you have any concerns about speed through the converter, we suggest measuring system performance before and after moving your instruments to the LAN/GPIB gateway.

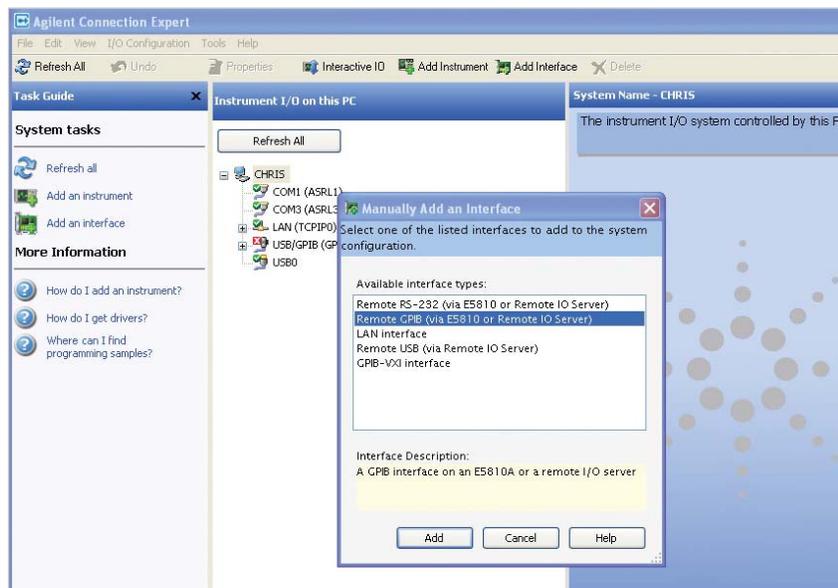


Figure 4. The Agilent Connection Expert utility can help you discover and connect GPIB instruments through the Agilent E5810A LAN/GPIB gateway.

## Step 5

### Analyze basic performance

At this point you should have a fully functional LAN- and LXI-based system. Next, it's time to determine if you are getting the performance you need.

In Steps 1 through 4, our objective was to get the system functioning on the LAN as quickly as possible. Now it is time to consider some selective tuning. We'll assume that you have tools to monitor where your program is spending time. Many programming languages have timer functions you can use to understand overall performance.

Since you've just migrated to a potentially faster instrument and a hybrid system, there are two common areas to investigate:

- **Wait statements**

Because the new instrument may be significantly faster than the unit it replaces, you may be able to shorten (or remove) any delays inserted into your existing code to account for the speed of the older instrument.

- **Compatibility mode slowdowns**

In Step 3, we used the VXI-11 compatibility mode to connect the new LXI instrument to the system. For any instrument that makes multiple fast measurements, consider recoding any critical looping routines to use a faster communication mode. When using VXI-11, for example, you can use the LAN sockets capability for some or all of your code. Referring back to Step 3, the LAN instrument was addressed as "TCPIP0::156.140.71.39::INSTR". You can also open a sockets port to that address using "TCPIP0::156.140.71.39::5025::SOCKET" (port 5025 is the sockets port for all Agilent instruments; the port number may be different for LXI instruments from other manufacturers).

Regarding the use of sockets, there are a few additional considerations. While Agilent has implemented virtually the entire GPIB command set in VXI-11, the sockets commands are instrument-dependent. Most instruments implement common functions such as reads, writes, triggers, and standard control functions, but are unlikely to implement GPIB-specific functions such as bus polling. In addition, use care when using binary reads to terminate by byte (or word) count: With binary data, early termination is possible if the binary data contains the same sequence as the termination character. (Note: VXI-11 and GPIB automatically handle this issue.)

More information on sockets and VXI-11 programming is available through the help system built into the Agilent IO libraries.

With those five steps completed, your system should be ready to go.

## Example 2 Maximizing overall performance

In this scenario, test speed is the top priority. Although you probably can't replace all of your existing test equipment, you can still build a hybrid system that achieves excellent performance.

This process is more situational than in Example 1: Key factors include the DUT, its test requirements, instrument choices, and your cost goals. In most cases, though, relatively simple choices can make a significant difference in overall system performance. The process described here will help you decide which tradeoffs will provide the performance you need.

### Determine where time is spent

System performance is dependent on every component of the test system. Even so, there is usually some portion of the test plan that tends to dominate overall test time. Applying the Pareto principle can help you understand where test time is being spent and then help you decide where to start. This requires some performance-analysis tools, and the easiest one is often the timer in your programming language.

Most hybrid test systems combine old and new test equipment. If a new instrument is significantly faster than the one it replaces, it may improve your overall test time. To make this determination, chart out the tasks that must be completed and define how you will accomplish them. An example DUT (Figure 5) will help illustrate this process. It has the following testing requirements:

- **DUT power-on and supply voltage**  
Initial power-on (normal operating) plus high, low and standby tests.
- **Inputs**  
Three inputs, which require two source setups at each supply voltage.
- **Outputs**  
Eight outputs that require voltage measurement. Each point must be scanned at each voltage level and with both source setups (except standby level).

With these requirements in hand, we can outline a suitable test sequence (Figure 6).

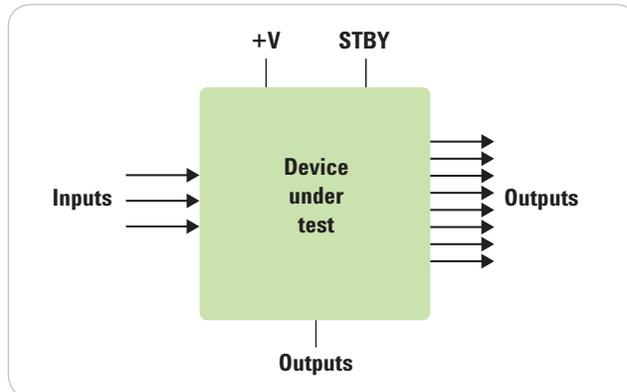


Figure 5. This simple device under test is the object of our example process.

#### Sequence:

```
Set power on (normal +V)
Source setup 1
Scan output (8 points)
Source setup 2
Scan output (8 points)
Set power voltage high
Scan output (8 points) – still source setup 2
Source setup 1
Scan output (8 points)
Set power voltage low
Scan output (8 points) – still source setup 1
Source setup 2
Scan output (8 points)
Set standby power
Source setup 1
Scan output for low state (8 points)
```

Figure 6. Taking a close look at your test procedure can reveal opportunities for speed improvement.

## Identify likely speed improvements

Note that the sequence in Figure 6 requires four voltage levels, five source setups and seven output scans of eight points each. Because the total number of output readings far exceeds the number of source setups and voltage changes, it appears the greatest opportunities for speed improvements are in the switching operations and the output measurements. It's also possible to make all measurements at one supply voltage before moving on to the next voltage.

Any measurement instruments in the receiver section would be top candidates for replacement if the newer model can provide faster measurement times. Additionally, it would be useful to seek improvements in switching speed, perhaps by changing to switch-and-measurement hardware that is combined in one unit designed for optimum speed. Second-tier priorities would be changes to the source setups and the various power supply voltage settings.

## Consider the computer connections

After examining the instrumentation, the next opportunity for speed improvement is in the choice of computer connection. If you are using VXI or PXI, a direct connection is best unless it is used only for switching or slow devices. For GPIB instruments, you can choose between an interface card and a converter such as the E5810A LAN/GPIB gateway. A direct interface card in the computer provides the fastest execution speed but limits your ability to remotely access the instruments.

You can estimate whether there will be a noticeable speed difference in using a LAN/GPIB converter. In this example, a direct interface card might be useful if switching or measurements are done via GPIB. However, the power supply and source (if GPIB) would have little impact on measurement times and either a card or converter would be OK.

## Tune system performance

A few general tuning tips—including a few that are applicable to any architecture—can help you get everything running at optimum speed. The following tips can be used in conjunction with your engineering judgment and experience.

### Basic tips

- Remove any instrument-reset commands from the main program flow. As a best practice, these should be used only during an initialization routine or when a significant operational error requires a restart from a known condition.
- Use quiescent times to calibrate instruments that perform auto-calibration routines. Doing this during, for example, shift or DUT changes will minimize lost time during active testing.
- Eliminate unnecessary instrument range changes as well as autoranging or autoscale operations because these can significantly slow down your measurements. As an alternative, try using the highest necessary range that won't require mid-test changes.
- Look for stray debug code that can be deleted from your test program.
- Simplify your program: remove intermediate display steps; simplify displays so less calculation occurs during critical measurements; simplify or remove monitoring checks; and move monitoring checks out of looping operations.

### Intermediate-level tips

- Manage your wait states. Where possible, try using the \*opc? (operation complete query) instead of wait statements. If a wait statement is necessary, use a measurement tool to determine exactly how long the delay must be. One useful tool is an oscilloscope, which can be inserted at key points to measure timing and help you determine actual performance. If you have replaced any part of the system (e.g., with a newer instrument), consider retuning any wait statements.

- Group similar measurements together. Changing from one instrument function to another tends to be relatively slow, so try to minimize the frequency of such changes. Similarly, if you keep track of the instrument states, you might be able to avoid unnecessary range or function changes.
- Turn off instrument screen updates. Instruments such as spectrum analyzers and oscilloscopes can expend a significant portion of their internal resources on screen updates. This ensures a high screen update rate for manual operation but may cause slower automated testing.
- Use a faster instrument. Your favorite (but old) DMM or analyzer might be costing you valuable test time—and it might be more useful on the bench. Modern DMMs can be 10 to 100 times faster for non-AC (or aperture-limited) measurements than the previous generation. New spectrum analyzers have much faster sweep times and wider resolution-bandwidth settings, which enables measurements without a long, slow full sweep.
- Examine your switching system. It may be possible to replace slow armature switches with faster reed relays. If you are performing high-impedance measurements, these might be able to use FET switches. For many test plans, switching is a major contributor to overall test execution time. If only a few pins must switch higher voltages or currents, using two small switch cards (e.g., one armature, one reed) instead of one larger card (all armature switches) might improve your test time.

### Other considerations

- Reassess your power supplies. Some of these are notoriously slow and basic functions such as DUT power-up and voltage changes can add significantly to overall execution time. Many modern power supplies (e.g., LXI) have much faster command processing and active downprogramming. With these features, units such as the Agilent N6700B modular power system can provide much faster high-to-low voltage transitions.
- Preset your stimulus. Downloading stimulus waveforms in advance and simply switching to a specific signal during a test can save a lot of execution time. Many of today's signal sources and function/arbitrary waveform generators provide storage for multiple preloaded waveforms.

### Accelerating LXI instruments

Most LXI instruments have additional functions and capabilities that can further improve your test times. For example, because many LXI instruments are “smart instruments,” they enable faster execution by offloading tasks such as computer processing or waveform pre-downloads.

Leverage built-in compatibility and speed: Many LXI instruments have both GPIB and LAN interfaces and can connect using either interface. Simply switching from GPIB to LAN can speed up tasks such as waveform downloads and data transfers.

Note that while the move from GPIB to LXI can often be made with a simple address change to your program, this may not be the instrument's fastest mode. Many LXI instruments—including most Agilent LXI instruments—provide a GPIB-compatible programming mode. Although this mode simplifies migration from GPIB, it adds an extra layer of software to provide compatibility at the expense of speed. The alternative is to use LAN sockets programming, which has less operational overhead and can significantly improve I/O performance (Figure 7).

### Close any web monitor windows

One key advantage offered by many LXI instruments is the ability to monitor the instrument during program execution. This is useful during debug but can impact execution time.

### Create a private network or subnetwork

This can be done by putting the instruments behind a LAN router. If your instruments are connected to a busy network, their performance will be affected by overall traffic levels. If you are using a relatively large subnet, try to limit the number of LAN switches in the path between the controlling computer and the instrumentation. We recommend this configuration for virtually all LAN-based test systems (refer to Agilent Application Note 1465-10 for specifics).

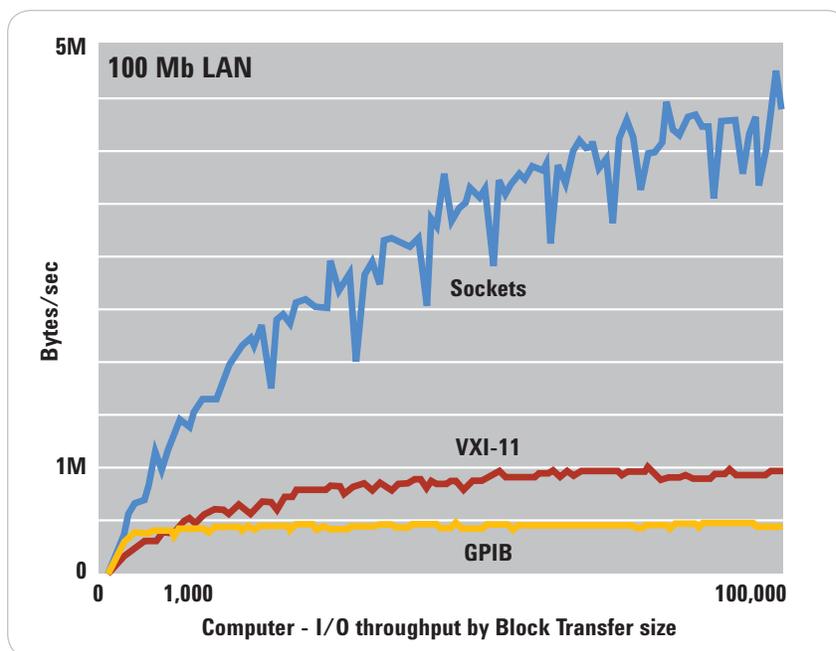


Figure 7. Sockets programming provides a significant speed advantage over GPIB and VXI-11.

### Download measurement routines into the instrument

Capabilities differ by instrument, but a number of LXI instruments allow preloading of test routines. For cases such as the DUT shown in Figure 5, switching and measurement routines could be preloaded into an Agilent 34980A switch/measure unit. The graphs in Figure 8 illustrate the level of speed improvement made possible by managing both switching and measurement within a tightly integrated instrument.

### Balance instrument vs. computer analysis

In some cases, using an instrument's on-board capabilities to perform data analysis and return the results will be faster than transferring a large data file to the host PC and using it to do processing and analysis. This type of preprocessing is possible with many of today's "smart" LXI instruments. Although the possible time savings are situation dependent, this technique may provide significantly shorter testing times.

### Use direct SCPI programming

Compared to an instrument driver, direct SCPI programming (using sockets) generally provides the fastest speeds possible with any instrument. This is true because many instrument drivers for LXI instruments are written in compatibility mode (using the VXI-11 standard) to enable easy migration from GPIB. While this provides simplified programming, it does not take full advantage of an instrument's native speed. Note that you can also use the instrument driver for a majority of functions while using direct SCPI programming for performance-sensitive measurements.

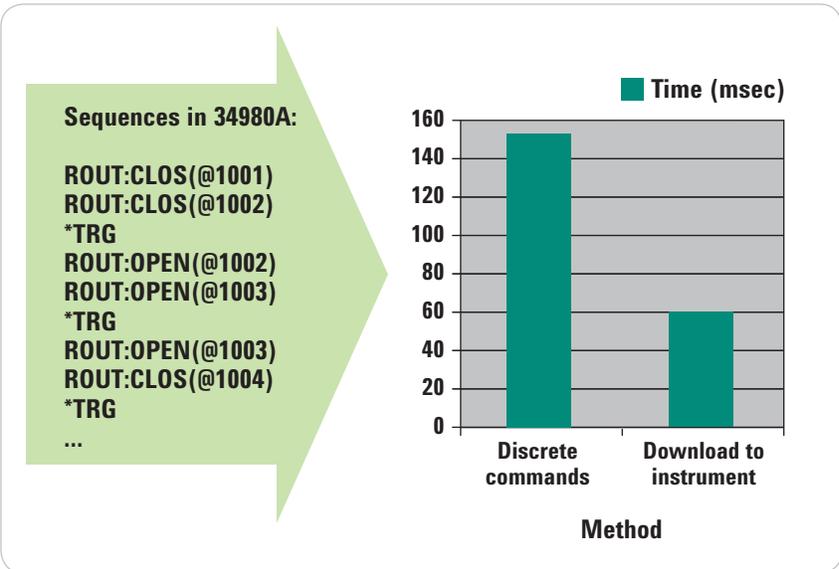


Figure 8. Internal management of switching and measurement within the 34980A switch/measure unit provides significant gains in speed.

### Check measurement speed and I/O speed

Either or both may be limiting the speed of your test routines. Because LAN is a serial interface with standard protocols (and a standard protocol stack) it has some inherent transaction latency. Although this can affect overall performance in transaction-intensive applications, overall execution time is more often dominated by factors such as measurement time, mode-switching, DUT settling, and switching operations.

### Add LXI Class B

The preceding tips apply to many of today's LXI Class C instruments. With Class B LXI instruments coming to market in 2008, you will be able to apply a host of new techniques—trigging, synchronization, time awareness—that will improve system performance. What's more, Class B will make both speed characterization and performance tuning much easier. These capabilities will be the subject of a future application note.

## Conclusion

Whether you seek to minimize development time or maximize system performance when creating a hybrid test system, the tips presented here can help ensure your success. The common thread running through either scenario is the use of LAN for I/O, the use of a LAN/GPIB gateway for older instruments, and the addition of one or more current-generation LXI instruments to the test system. The speed, flexibility and capabilities provided by LAN and LXI can provide significant reductions in system development time and measurement execution time.

## Related Agilent literature

The 1465 series of application notes provides a wealth of information about the creation of test systems, the successful use of LAN, WLAN and USB in those systems, and the optimization and enhancement of RF/microwave test systems.

- *Test-System Development Guide: A Comprehensive Handbook for Test Engineers*  
(pub no. 5989-5367EN)  
<http://cp.literature.agilent.com/litweb/pdf/5989-5367EN.pdf>

### Test System Development

- *Test System Development Guide: Application Notes 1465-1 through 1465-8*  
(pub no. 5989-2178EN)  
<http://cp.literature.agilent.com/litweb/pdf/5989-2178EN.pdf>
- *Using LAN in Test Systems: The Basics*  
AN 1465-9 (pub no. 5989-1412EN)  
<http://cp.literature.agilent.com/litweb/pdf/5989-1412EN.pdf>
- *Using LAN in Test Systems: Network Configuration*  
AN 1465-10 (pub no. 5989-1413EN)  
<http://cp.literature.agilent.com/litweb/pdf/5989-1413EN.pdf>
- *Using LAN in Test Systems: PC Configuration*  
AN 1465-11 (pub no. 5989-1415EN)  
<http://cp.literature.agilent.com/litweb/pdf/5989-1415EN.pdf>
- *Using USB in the Test and Measurement Environment*  
AN 1465-12 (pub no. 5989-1417EN)  
<http://cp.literature.agilent.com/litweb/pdf/5989-1417EN.pdf>
- *Using SCPI and Direct I/O vs. Drivers*  
AN 1465-13 (pub no. 5989-1414EN)  
<http://cp.literature.agilent.com/litweb/pdf/5989-1414EN.pdf>
- *Using LAN in Test Systems: Applications*  
AN 1465-14 (pub no. 5989-1416EN)  
<http://cp.literature.agilent.com/litweb/pdf/5989-1416EN.pdf>

- *Using LAN in Test Systems: Setting Up System I/O*  
AN 1465-15 (pub no. 5989-2409EN)  
<http://cp.literature.agilent.com/litweb/pdf/5989-2409EN.pdf>
- *Next-Generation Test Systems: Advancing the Vision with LXI*  
AN 1465-16 (pub no. 5989-2802EN)  
<http://cp.literature.agilent.com/litweb/pdf/5989-2802EN.pdf>

### RF and Microwave Test Systems

- *Optimizing the Elements of an RF/Microwave Test System*  
AN 1465-17 (pub no. 5989-3321EN)  
<http://cp.literature.agilent.com/litweb/pdf/5989-3321EN.pdf>
- *6 Hints for Enhancing Measurement Integrity in RF/Microwave Test Systems*  
AN 1465-18 (pub no. 5989-3322EN)  
<http://cp.literature.agilent.com/litweb/pdf/5989-3322EN.pdf>
- *Calibrating Signal Paths in RF/Microwave Test Systems*  
AN 1465-19 (pub no. 5989-3323EN)  
<http://cp.literature.agilent.com/litweb/pdf/5989-3323EN.pdf>

### LAN eXtensions for Instrumentation (LXI)

- *LXI: Going Beyond GPIB, PXI and VXI*  
AN 1465-20 (pub no. 5989-4371)  
<http://cp.literature.agilent.com/litweb/pdf/5989-4371EN.pdf>
- *10 Good Reasons to Switch to LXI*  
AN 1465-21 (pub no. 5989-4372EN)  
<http://cp.literature.agilent.com/litweb/pdf/5989-4372EN.pdf>
- *Transitioning from GPIB to LXI*  
AN 1465-22 (pub no. 5989-4373EN)  
<http://cp.literature.agilent.com/litweb/pdf/5989-4373EN.pdf>
- *Creating Hybrid Systems with PXI, VXI and LXI*  
AN 1465-23 (pub no. 5989-4374EN)  
<http://cp.literature.agilent.com/litweb/pdf/5989-4374EN.pdf>
- *Using Synthetic Instruments in Your Test System*  
AN 1465-24 (pub no. 5989-4375EN)  
<http://cp.literature.agilent.com/litweb/pdf/5989-4375EN.pdf>

- *Migrating System Software from GPIB to LAN/LXI*  
AN 1465-25 (pub no. 5989-4376EN)  
<http://cp.literature.agilent.com/litweb/pdf/5989-4376EN.pdf>
- *Modifying a GPIB System to Include LAN/LXI*  
AN 1465-26 (pub no. 5989-6824EN)  
<http://cp.literature.agilent.com/litweb/pdf/5989-6824EN.pdf>

### Using Linux in Your Test Systems

Example code is available for download at <http://www.agilent.com/find/linux>.

- *Using Linux in Your Test Systems: Linux Basics*  
AN 1465-27 (pub no. 5989-6715EN)  
<http://cp.literature.agilent.com/litweb/pdf/5989-6715EN.pdf>
- *Using Linux to Control LXI Instruments Through VXI-11*  
AN 1465-28 (pub no. 5989-6716EN)  
<http://cp.literature.agilent.com/litweb/pdf/5989-6716EN.pdf>
- *Using Linux to Control LXI Instruments Through TCP*  
AN 1465-29 (pub no. 5989-6717EN)  
<http://cp.literature.agilent.com/litweb/pdf/5989-6717EN.pdf>
- *Using Linux to Control USB Instruments*  
AN 1465-30 (pub no. 5989-6718EN)  
<http://cp.literature.agilent.com/litweb/pdf/5989-6718EN.pdf>
- *Tips for Optimizing Test System Performance in Linux Soft Real-Time Applications*  
AN 1465-31 (pub no. 5989-6719EN)  
<http://cp.literature.agilent.com/litweb/pdf/5989-6719EN.pdf>

### Building Hybrid Test Systems

- *Laying the Groundwork for a Successful Transition*  
AN 1465-32 (pub no. 5989-8175EN)  
<http://cp.literature.agilent.com/litweb/pdf/5989-8175EN.pdf>

[www.agilent.com/find/open](http://www.agilent.com/find/open)



### Agilent Email Updates

[www.agilent.com/find/emailupdates](http://www.agilent.com/find/emailupdates)

Get the latest information on the products and applications you select.



### Agilent Direct

[www.agilent.com/find/agilentdirect](http://www.agilent.com/find/agilentdirect)

Quickly choose and use your test equipment solutions with confidence.

## Remove all doubt

Our repair and calibration services will get your equipment back to you, performing like new, when promised. You will get full value out of your Agilent equipment throughout its lifetime. Your equipment will be serviced by Agilent-trained technicians using the latest factory calibration procedures, automated repair diagnostics and genuine parts. You will always have the utmost confidence in your measurements.

Agilent offers a wide range of additional expert test and measurement services for your equipment, including initial start-up assistance onsite education and training, as well as design, system integration, and project management.

For more information on repair and calibration services, go to

[www.agilent.com/find/removealldoubt](http://www.agilent.com/find/removealldoubt)

## www.agilent.com

For more information on Agilent Technologies' products, applications or services, please contact your local Agilent office. The complete list is available at: [www.agilent.com/find/contactus](http://www.agilent.com/find/contactus)

### Americas

Canada	877 894 4414
Latin America	305 269 7500
United States	800 829 4444

### Asia Pacific

Australia	1 800 629 485
China	800 810 0189
Hong Kong	800 938 693
India	1 800 112 929
Japan	81 426 56 7832
Korea	080 769 0800
Malaysia	1 800 888 848
Singapore	1 800 375 8100
Taiwan	0800 047 866
Thailand	1 800 226 008

### Europe & Middle East

Austria	0820 87 44 11
Belgium	32 (0) 2 404 93 40
Denmark	45 70 13 15 15
Finland	358 (0) 10 855 2100
France	0825 010 700*
	*0.125 €/minute
Germany	01805 24 6333*
	*0.14 €/minute
Ireland	1890 924 204
Israel	972 3 9288 504/544
Italy	39 02 92 60 8484
Netherlands	31 (0) 20 547 2111
Spain	34 (91) 631 3300
Sweden	0200-88 22 55
Switzerland	0800 80 53 53
United Kingdom	44 (0) 118 9276201

Other European Countries:

[www.agilent.com/find/contactus](http://www.agilent.com/find/contactus)

Revised: March 27, 2008

Microsoft, Windows and Windows NT are U.S. registered trademarks of Microsoft Corporation. Visual Studio is a registered trademark of Microsoft Corporation in the United States and/or other countries.

Product specifications and descriptions in this document subject to change without notice.

© Agilent Technologies, Inc. 2008

Printed in USA, May 19, 2008

5989-8176EN



Agilent Technologies