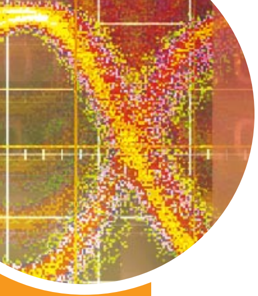


Don MacDonald, Agilent Technologies Inc.; Tim Anderson, Motorola Computer Group; Mark Overgaard, Pigeon Point Systems

Windows-Based Systems Can Provide High Availability For Telecommunications Applications

Windows and CompactPCI team up for high-availability systems.

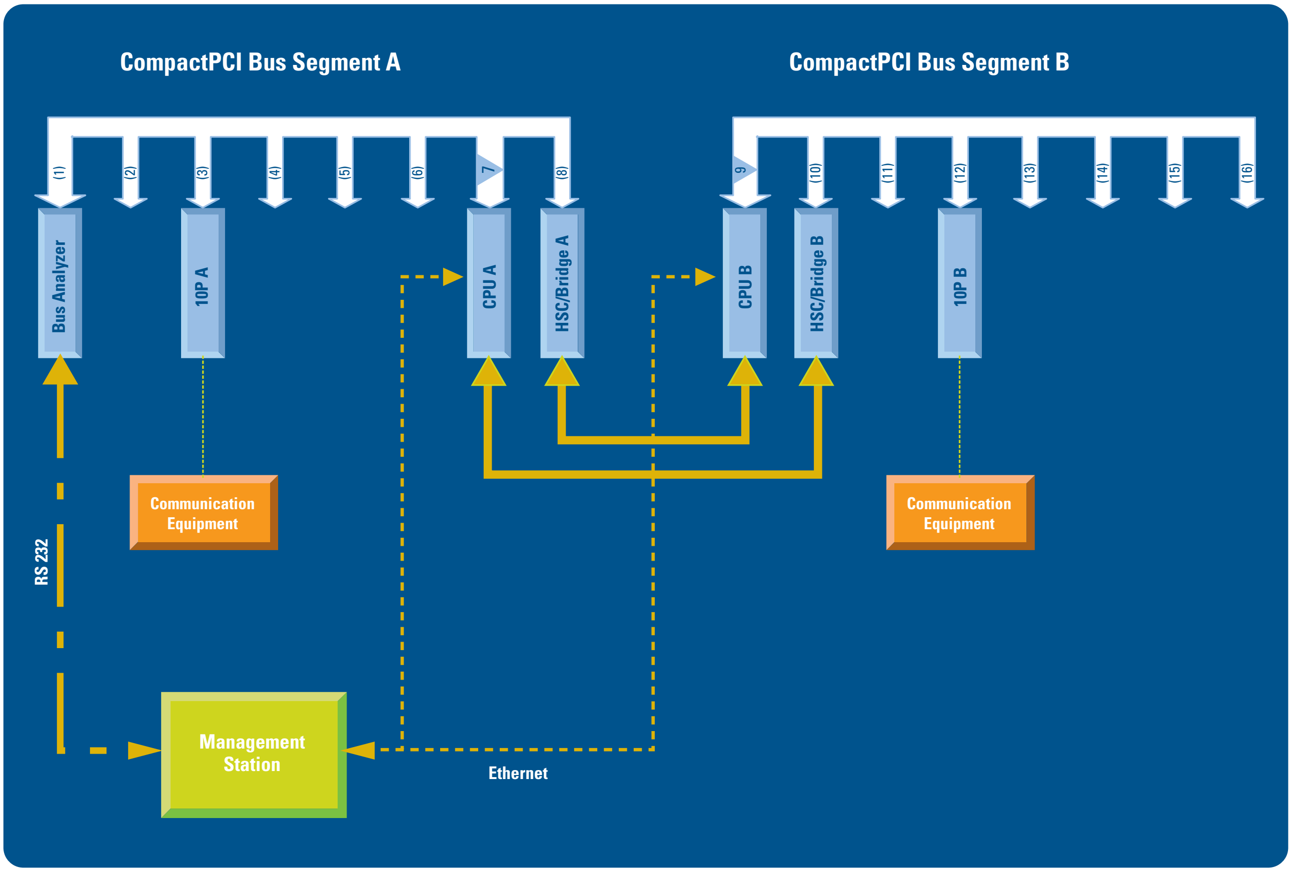


Figure 1. The hardware configuration used to demonstrate a High-Availability CompactPCI implementation.

The degree to which our information-driven society relies on long-distance communication makes uninterrupted telecommunications imperative. Everyone from a store clerk ringing up a credit-card transaction to an air traffic controller trying to route jumbo jets around dangerous weather has come to rely on instantly available telecommunications service.

To provide this level of service, telecommunications providers require extremely reliable equipment at the nodes of their networks. These high-availability (HA) systems must achieve uptime performance of at least “five nines”— they must be operating 99.999 percent of the time, allowing only 5 minutes of downtime per year. If there are 10 component failures over a year, each has a budget of less than 30 seconds for recovery!

Of course, it is impossible to diagnose and repair a piece of equipment in that time, especially at remote, automated installations where it may take repair technicians hours to reach the equipment location before they can even start making repairs. The only option is to include redundant functional modules and the means to automatically sense failures in one module and reroute its traffic load to backup equipment. In the past, the equipment available for this service has used proprietary embedded-control systems to coordinate the operation of the redundant components. This approach makes telecommunications equipment very expensive and lengthens the development process.

A Less Expensive Approach

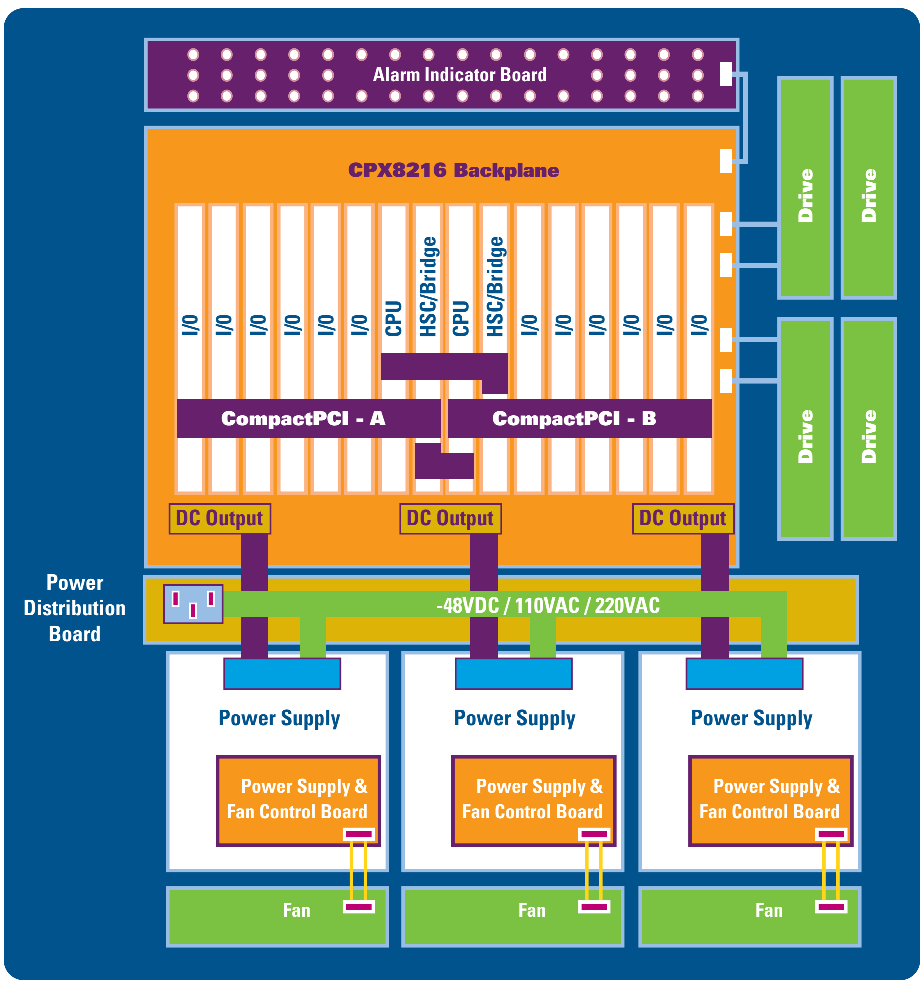
A much less expensive approach would be to make maximum use of commercial off-the-shelf hardware and software components to build systems with comparable functionality and reliability. Over the past three years, members of the PCI Industrial Computer Manufacturers Group (PICMG) have aggressively pursued the goal of building HA telecommunications systems using generally available industry-standard computer technology.

Recently we have demonstrated automated rapid "fail over" between redundant telecommunication system resources, including I/O boards and processor boards. **Figure 1** shows the hardware configuration of the demonstration system. It begins with a CompactPCI backplane consisting of two bus segments. Each segment has its own control processor module (CPU), an I/O processor module (IOP), and a Hot Swap Controller (HSC)/bridge module. In one segment there is also a CompactPCI bus analyzer/exerciser module.

The CompactPCI system represents a remote communications facility functioning as a network node. It could be installed at the base of a remote cell-site tower on the top of a mountain, or at some other hard-to-reach location. This remote facility would communicate with a management station via a TCP/IP network, embodied in the demonstration system by an Ethernet connecting the two CPU modules to the management station.

In a real cell-site facility, the IOPs would control the operation of the radio receivers and transmitters that carry the communications traffic. In the demonstration system, these are simulated by the IOP modules and their sample software.

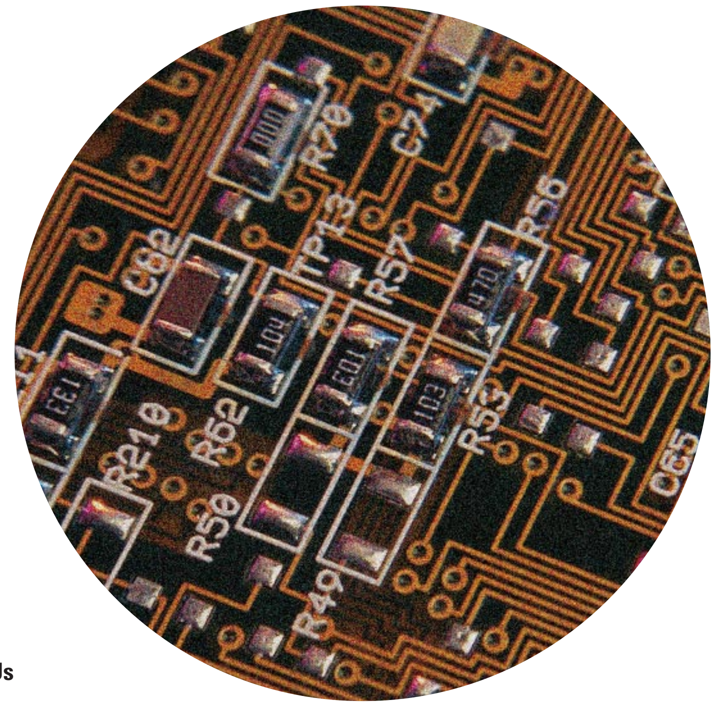
Figure 2. The HSC/bridge modules link the two bus segments and make CPU redundancy possible.



The demonstration system hardware is a Motorola Computer Group CPX8216 High Availability Compact PCI computer platform. The CPX8216 provides two eight-slot CompactPCI bus segments, two CPU slots, and two slots for the HSC/bridge modules. The remaining slots can be used for general-purpose I/O cards such as the IOPs. In addition, the CPX8216 provides n+1 redundant power supplies, hot-swap disk drive modules, and a system-status alarm panel. The demonstration system uses two IOPs, one in each segment.

The bus analyzer/exerciser module is an HP E2940A CompactPCI bus analyzer/exerciser from Agilent Technologies, which has a fully programmable PCI master and target, plus onboard memory. It provides visibility into the bus traffic, and monitors the transfer of control. Of course there would be no need to leave a bus analyzer/exerciser permanently installed in every remote facility. There would be times, however, when bus-analysis capabilities would be useful for diagnostic or optimizing procedures. The module's hot-swap capability would make it quite simple for a technician to carry a unit out to the site, plug it in, perform the needed tests, and pull it back out without interrupting the communications traffic flow.

The CPU modules contain Pentium II microprocessors running Windows 2000 (Win2K). They provide system management functions such as bus arbitration. Real-time control functions for the radio equipment are provided by the IOPs, which have their own in-module processors (PowerPC 750s) running the LynxOS real-time operating system (RTOS) from Lynx Real-Time Systems.

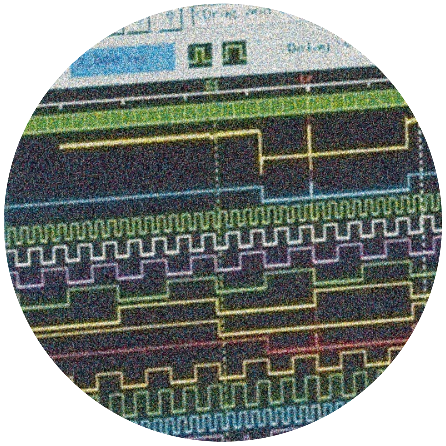


Bridge Between CPUs

The HSC/bridge modules link the two bus segments and make CPU redundancy possible **(Figure 2)**. The CPX8216 hardware is capable of operating the two CompactPCI buses in several different modes. The system can operate with only one CPU running both bus segments. Another configuration has two CPUs each controlling a bus segment. The demonstration system runs with one CPU controlling both segments and the second CPU in a standby role, ready to take over both segments if necessary. The Hot Swap Controller function on each bridge module adds radial control by the corresponding CPU of the key attributes of each slot in the system (as detailed below).

Figure 3 shows the software components needed to perform these functions. They all run the Windows 2000 operating system. Under Windows, Microsoft's Windows Management Infrastructure (WMI) collects status and events summarizing the operation of the system, to be used at the separate management station for monitoring purposes. The Hot Swap Kit (HSK) and Hot Swap Controller Kit (HSCK) products from Pigeon Point Systems augment the standard facilities of Windows 2000 for high-availability CompactPCI systems. The HSK, which is designed to work on any CompactPCI hot-swap system, leverages the Windows plug-and-play facility to support hot swapping of CompactPCI boards, including assigning and releasing resources, and loading and removing drivers. The HSCK layers on top of the HSK to support the advanced features of the CPX8216, including its individual slot controls and dual redundant CPU facility.

Finally, there is the backplane messaging software, also from Pigeon Point Systems, which allows the CPUs and IOPs to communicate with each other over the bus. The IOPs run the LynxOS RTOS and backplane messaging software. Not shown is the application software needed by the IOPs to control the communications equipment in a real system, but which is not needed for the demonstration.



The management station is a separate unit from the CompactPCI system, and implements the system manager's control functions. It is an ordinary PC-based workstation running Windows NT. Microsoft's Management Console (MMC) software runs under Windows NT, and allows the system manager (or, in the case of the demo, the engineer presenting the system) to view system status and operation, and to make changes. The MMC contains a CompactPCI snap-in, which is the software that actually communicates over the Ethernet with the WMI instances on the two CPU modules.

Being a general-purpose workstation, the management station also can perform other functions. In the demonstration system, the management station runs an application program interface (API) for the Agilent bus analyzer/exerciser, allowing users to set parameters and obtain reports from its activities.

Putting High Availability to the Test

Clearly, there are two concerns when a module connected to the bus fails: maintaining communications services, and protecting the integrity of the bus and its connected modules.

To protect the bus, the offending module must be effectively disconnected from the bus. Since the whole operation must be done automatically and virtually instantaneously, physically removing the module from the chassis is not an option.

High-availability CompactPCI platforms, such as the CPX8216, are able to cut the module's power input and isolate the signal lines under system software control via the Hot Swap Controller. A hot-swappable CompactPCI module is required to respond to a backplane signal called BDSEL (board select). When that signal is deasserted by the HSC, the module must turn off the bulk of its power. Also, its backplane interface, which drives all of the signal-line connections, goes into a high-impedance tristate, so that the module is no longer an active member of the backplane. Thus, the module is electrically isolated from the backplane, even though it is still physically plugged into it.

There are two sets of power pins going to the board. The bulk of the power is called "back-end" power. It powers most of the board behind the backplane interface. That power is turned off as a consequence of deasserting BDSEL. The "front-end" power remains on to make sure that the bus interface remains in tristate.

Maintaining communication services in the face of component failure is much more difficult. Suppose one of the IOPs fails. The first problem is to recognize that there is a problem. One means of signaling a problem is the HEALTHY signal. A module asserts HEALTHY to communicate that its power is good and that it passes its internal self tests. Deasserting HEALTHY signals that the board is in trouble. All Compact PCI boards are required to implement a HEALTHY signal that at least confirms that power is good, but only high-availability platforms have the means to monitor HEALTHY explicitly. When the CPU sees a module's HEALTHY line deasserted (or if it detects some other sign of ill health, such as loss of a special "heartbeat" signal in the backplane messaging layer), it marks that IOP out of service and reroutes its traffic to other IOPs.

Having a CPU go down is somewhat more problematical. For a system operating in an Active/Standby configuration, such as the demo system, recovering requires the standby CPU to configure the bridge module to take control of both CompactPCI bus segments. In an actual failure, this would be done locally on the system and reported to the management station. In the demo system, the "active" and "standby" CPUs are asked to do a cooperative handover by the management station.

The Hot Swap Controller Kit provides APIs for managing the cooperative handovers between the active and standby CPUs, as well as for monitoring the HEALTHY signal and controlling the BDSEL signal for each slot in the system.

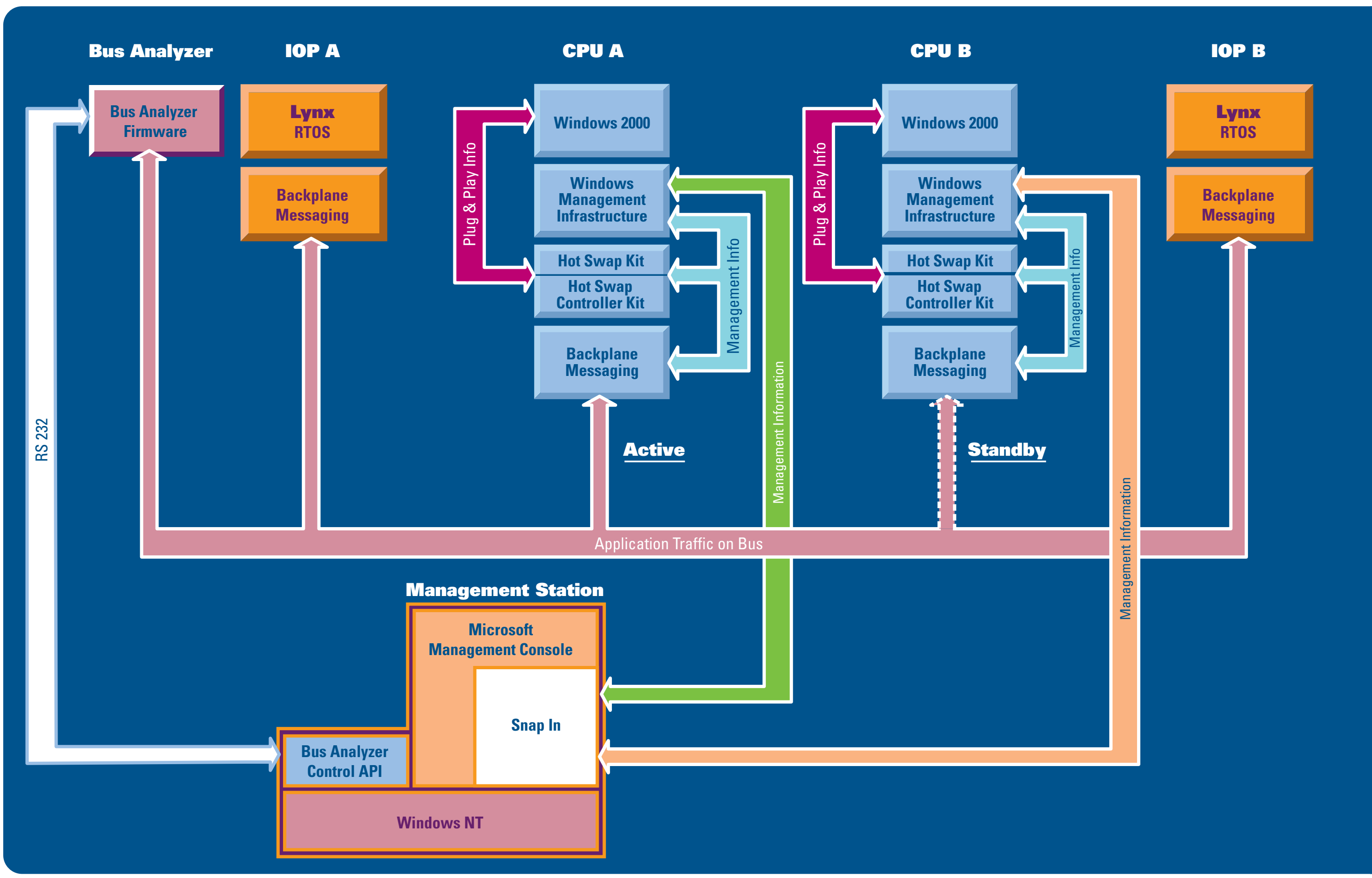
In a real-life situation such a cellular phone call, a node failure needs to be detected and recovered from very quickly. If an active CPU fails, the standby CPU needs to be ready to take the system and understand the operations that were being performed by the active CPU. The detection of failure then starts the switchover. The IOPs are added into the already-running standby processor and configured. The IOP then can continue its data transfer, picking up at the last known good point. The CPU switchover and system reconfiguration should be quick enough so that little if any change is noticeable, and phone calls in progress continue with little if any interruptions.

Analyzer Activities

The Agilent HP E2940A CompactPCI bus analyzer/exerciser provides a convenient way to exercise many of a high-availability system's functions during the system integration phase of product development. We wanted the system to run on an automated basis, but we also didn't want to write C code for invoking all the functions. Instead, we integrated the needed API functions within a FORTH-language wrapper that runs on the management station. That allowed us to easily write control code using FORTH that was invoked from the Windows command line on the management station.

The bus analyzer/exerciser monitors all messages sent across the bus during the CPU transfer. It has built-in firmware for performing some 60 to 70 protocol checks on every bus cycle. In the demonstration system, most of the bus analyzer/exerciser's checks are used, but a few are turned off. For example, when a board flags an interrupt to the central system, it

Figure 3. The software components required for high-availability systems.



asserts a line on the bus. When the bus analyzer/exerciser sees that flag, it starts a timer. If it doesn't see an interrupt acknowledge signal (IACK) come back within a preset number of bus cycles, then it logs an error. However, in the demonstration system there are bridges between the physical slots in which the boards reside and the CPU. The interrupt information goes past the bridges to the CPU, but there is no IACK signal that comes back. If the bus analyzer/exerciser were plugged into a conventional system, the absence of an IACK response could well be something to worry about. In this system, however, there is no IACK requirement, so the IACK error does not signal a real problem and is marked as irrelevant.

Incorporating the CompactPCI bus analyzer/exerciser into the demonstration system helped a great deal while doing hot-swap and dual-host system development. It provided visibility into the bus that would otherwise be unavailable. With the C-language API and the FORTH wrapper, we were able to very easily set up the unit for different types of monitoring. This bus analyzer/exerciser is an excellent example of technology developed originally for the broader PCI market but applicable (with extensions such as hot-swap support) to the CompactPCI domain. Also, we were able to leverage the plug-and-play facilities of a high-volume operating system (Windows 2000) and the included management instrumentation to support the very specialized needs of high-availability CompactPCI systems.

Our demonstration system shows that CompactPCI platforms provide a fine foundation for building lower cost high-availability systems for telecommunications applications.

Analyzer/Exerciser Eases the Evaluation of Compact PCI for High-Availability Applications

The Agilent Technologies HP E2940A CompactPCI bus analyzer/exerciser has many capabilities to assist you in the development of high-availability telecommunications and other CompactPCI systems. It has recently been evaluated to run at exerciser speeds of 66 MHz, in addition to the 66-MHz analyzer speed previously specified.

There are three ways in which the bus analyzer/exerciser can be controlled. One is through a set of C-language API functions. Another is a graphical user interface (GUI). There is also a command-line interface, a one-line window in the GUI that allows you to interactively key in shorthand versions of the commands that you want the bus analyzer/exerciser to accept.

The CompactPCI bus analyzer/exerciser provides a convenient way to exercise many of a high-availability system's functions during the system integration phase of product development. The analyzer/exerciser tool is a real PCI device, with a real configuration space header and a set of target decoders for emulation of any PCI device. For example, you can establish what the bus analyzer/exerciser emulates on a Compact-PCI bus by programmatically (or manually) writing device/vendor ID, requested memory/IO space size, device class, and other information into its configuration registers. Plug-and-play operating systems (like Windows 2000) use these blocks of information to automatically determine what drivers to load. As a result, you can cycle through a whole series of identified and designed board personalities to ensure that your operating system is capable of coping with the boards you expect to be present in your mainframe.

Another useful feature of the CompactPCI bus analyzer/exerciser for anyone developing products that incorporate this Windows high-availability technology is its hot-swap capability. This enables you to insert and extract it while the system continues to run. Consider that the system is up and working, but not optimally. You want to understand what is going on and diagnose the problem without affecting the system's ability to continue delivering service. Hot swapping allows you to insert the bus analyzer/exerciser onto the bus and get a view of system behavior without degrading system availability.

For more information, check 5 on the reply card, or visit http://www.agilent.com/find/insight6.