Designing a Vibration Measurement Application With LXI

by Kristin Sullivan, Data Translation

New cell phones, PDAs, Netbooks, and fancy consumer devices are being introduced at breakneck pace. These developments must get to market quickly, and assuring their integrity requires that severe shock and vibration testing is done faster and better than ever before. Time is of the essence to get large quantities of new products in the consumer's hands. Without adequate drop tests, shock-mode dynamics, and force measurements, customer acceptance may be compromised.

The emergence of the LXI standard holds the promise of plugging together off-the-shelf measurement boxes that can quickly accomplish these vibration tests. Compatibility in software and hardware with the known experience of Ethernet, the basis of LXI, makes system tests feasible under the rapid time pressures for these new consumer products.

The Design Process

The trade-offs to make a flexible, high-accuracy vibration capability are many. To this end, a ready-to-use test resource completely LXI compatible was our objective when developing the DT8837 Instrument Module, an off-the-shelf solution. The design process began with three conceptual Ethernet architectures and their pros and cons.

First Approach: Using a Sophisticated Memory Controller

The first approach we considered implemented an Ethernet module connected through a serial interface to a USB processor and a separate large memory buffer to boost the amount of memory that would be available for in-module data storage. A sophisticated controller was required to make the memory appear to be a FIFO to the USB processor as well as to the incoming and outgoing data.

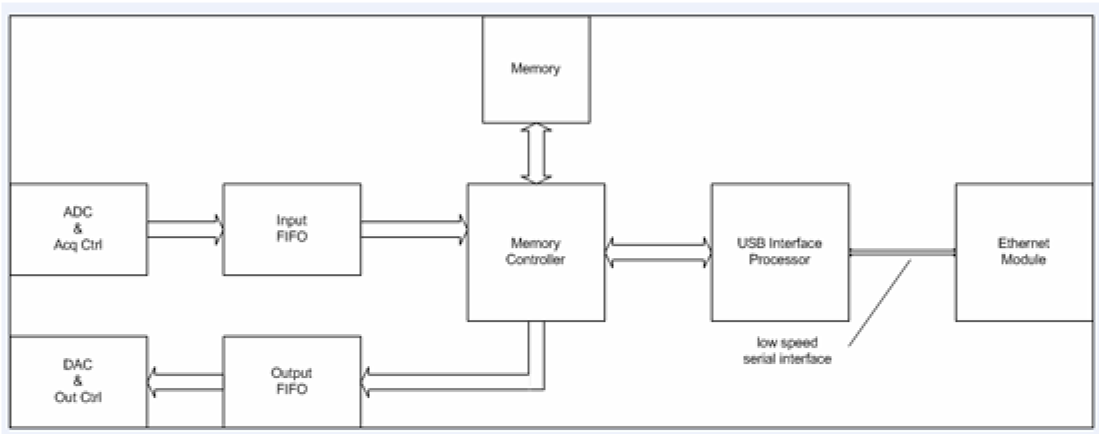

Figure 1. Block Diagram of Sophisticated Memory Controller Design Approach

The bottleneck here was the serial connection between the USB processor and the Ethernet module. Moving large volumes of data back to a host over the Ethernet connection would be slow; there were doubts that we could achieve the bandwidth of the 10/100 supported device.

A significant benefit of this design was our familiarity with the approach. Some recycling of prior designs could lead to a faster implementation, which, in turn, would get us to market faster. However, the time to design and debug the memory controller could negate this gain. Also, data transfer back to the host would be slow, additional memory would be expensive, and the USB processor would be inefficient for processing data.

Second Approach: Using a High-End Embedded Processor To Move Data

The second approach used a high-end embedded processor in place of the USB processor and Ethernet module. It removed some of the duplication of the first design but put the entire firmware burden on the processor. The processor core in this device would be quite capable of performing any of the tasks necessary, an improvement over the first design idea. This approach utilized an internal PCI bus to move data through the embedded processor to a memory buffer shared by input/output data and processor instruction and data.

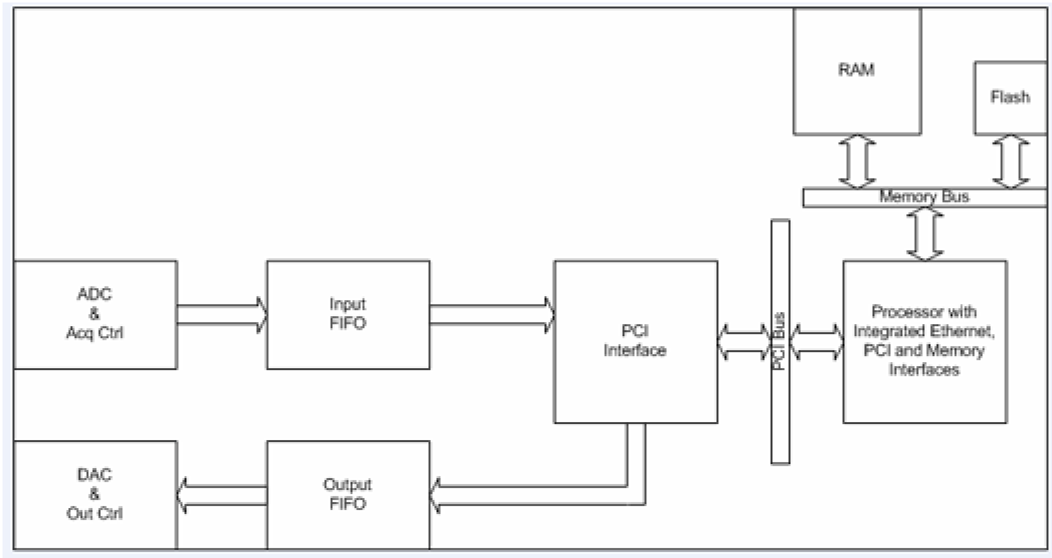

**Figure 2. Block Diagram of a High-End Processor
With PCI Bus Design Approach**

The pros of this design included faster data movement from the module resulting in improved user experience, a memory controller that was built into the processor, a platform that could be used and extended in future projects, removal of unnecessary memory duplication, and use of in-house PCI experience. It also would fully utilize the 32-bit processor core for any necessary data processing.

There were three major drawbacks to this approach: time to design and debug the PCI interface, the learning curve of a new processor and the associated tools, and the cost of the processor and PCI interface.

Third Approach: Using a System on a Module

The third approach used an Ethernet module similar to that in the first approach. But this one also included a parallel bus and a DMA engine to assist in data movement to and from memory. This system on a module incorporated an ARM7 processor with flash and DRAM and the added bus interface that the Ethernet module in the first solution did not have.

This solved the data transfer rate issues of the first solution while eliminating the large memory buffer and memory controller. If needed, a controller and more memory could be added to this solution to provide higher bandwidth in the future.

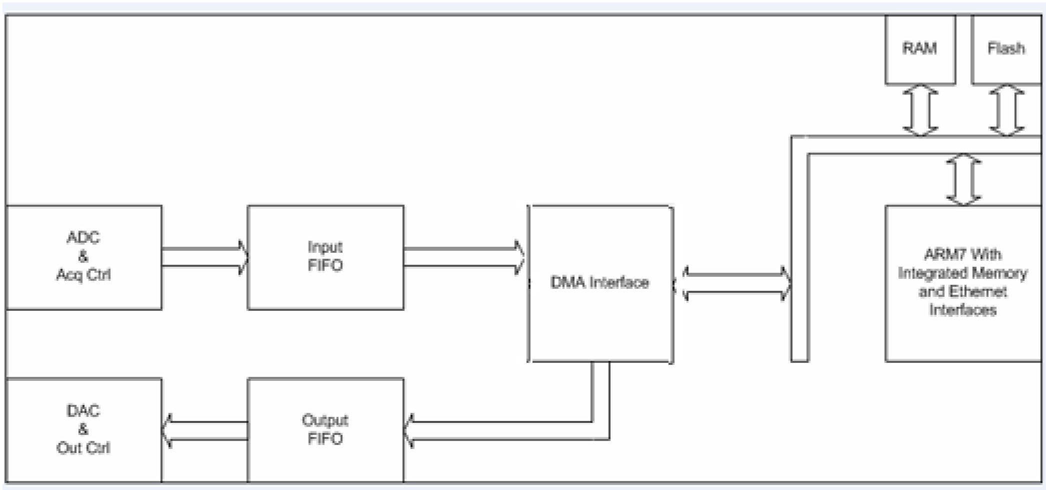

Figure 3. Block Diagram Implementing System on a Module Design Approach

Ultimately, the third approach made the most sense for a variety of reasons. It offered a path to the future with less risk, it was less expensive with reduced components, and most importantly, it provided the best user experience. The only deterrent to this design was learning a new processor.

Other Design Considerations

To produce a product with the utmost accuracy in noisy industrial environments, isolation was a key consideration. A common source of error in industrial measurement applications is introduced by noise and ground loops. The implied assumption is that each sensor's ground is at the same potential when, in reality, this may not be the case.

Factors that can contribute to these ground differences are long wiring runs, crosstalk from motors or generators, or high signal source impedance. To prevent this noise from entering the signal path, the signal must be isolated on a channel-to-channel basis as well as from the PC ground reference.

Our design implemented ISO-Channel™ technology to eliminate connection problems by using galvanic isolation methods to guarantee 1,000-V isolation between sensor grounds and ±500 V to earth ground. Each signal input channel has its own 24-bit ADC. As a result, accuracy is preserved for all sensor inputs.

In the next stage of the design, we implemented a feature of the LXI standard we knew would be important to remote vibration applications: wired trigger bus (WTB). This capability allows up to 16 LXI devices to be synchronized and clocked simultaneously in a centralized test stand. In addition to WTB features, support for

LAN packet triggering is included to allow devices that are not physically connected to be triggered simultaneously.

While the LXI standard does not dictate use of any particular software protocol, we standardized on SCPI. This allows new devices to be used easily in existing test fixtures that may feature non-LXI devices.

Monitoring devices in difficult or harmful conditions is of paramount concern with remote applications. The ability to share information or device controls with colleagues in any part of the world is fast becoming a necessity.

This remote accessibility is provided through mandated discovery utilities and an instrument Web interface. For example, Data Translation's Eureka Discovery Utility helps locate all LXI instruments connected to your system and provides the instrument's IP address, manufacturer, model number, serial number, and version of the firmware. You also can use this utility to configure the Windows firewall settings and update the firmware for your LXI instrument. The built-in Web interface allows you to verify the operation of your instrument and perform basic functions with Internet Explorer with no additional software. Using the Web interface, you can configure the instrument, control and measure input signals, and save results to disk.

An IVI-COM driver is another valuable software component required for LXI conformance. This driver provides a COM-based programming interface to access instrument functions. The IVI-COM driver works in any development environment that supports COM programming, including Data Translation's Measure Foundry, The MathWorks MATLAB, C#, VB.NET, Visual Basic, Agilent VEE Pro, and National Instruments' LabVIEW.

Conclusion

Using all the criteria for LXI compliance as a design guide, the result is an off-the-shelf product ready to plug-and-play with other instruments. By using a standards-based approach, you can be confident that this product will integrate with your existing systems and meet future demands as well.

About the Author

Kristin Sullivan is a product marketing manager at Data Translation. In six years with the company, she also has held positions in sales and technical support. Ms. Sullivan attended the University of Colorado. Data Translation, 100 Locke Dr., Marlboro, MA 01752, 508-481-3700, e-mail: ksullivan@datx.com