Increase Throughput with Parallel Test Technologies

Test system architects must interpret evolving technologies, such as multicore processors and high-speed data buses, as they design, develop, and implement test systems.

Frank Lloyd Wright, an influential 20th century architect, once said, "Every great architect … must be a great original interpreter of his time, his day, his age." The same can be said for test system architects. They must interpret evolving technologies, such as multicore processors and high-speed data buses like PCI Express, as they design, develop, and implement the systems they have been commissioned to build. By combining these technologies with NI LabVIEW software and NI TestStand test management software, test engineers can create high-performance test systems capable of parallel processing, parallel measurements, and even completely parallel test on the production floor.

Parallel Processing

In traditional CPU designs, performance is limited by pragmatic challenges such as heat dissipation because of high clock rates. To ensure the PC platform continues to keep pace with increasing processing demands, chip manufacturers are developing new processors with multiple processing cores. For automated test applications to realize the performance and throughput benefits of multicore technology, software applications must target the multiple processing cores by creating multiple threads that execute on the processing cores.

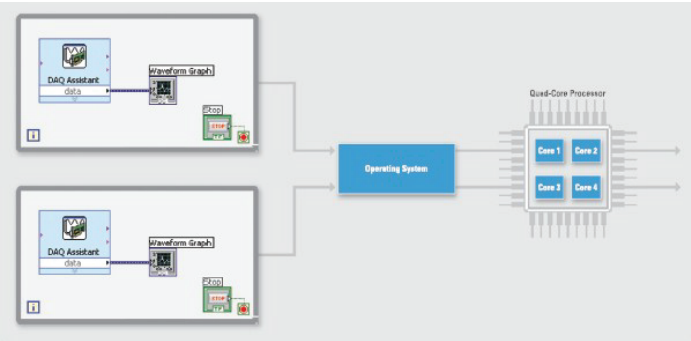

Figure 1. The LabVIEW compiler creates separate execution threads for parallel sections of code, with no user configuration required.

However, writing multithreaded applications in a text-based programming language, such as C, is nontrivial for most design and test engineers and requires expertise in the semantics of creating and managing the threads and passing data among them in a thread-safe way. Using graphical programming environments, such as LabVIEW, engineers can take full advantage of multicore processing power. As shown in Figure 1, two loops in LabVIEW that do not share a data dependency automatically execute in separate threads while abstracting the details of thread management away from the developer.

Parallel Measurements

Parallel measurements require each of the test system's subcomponents, not just the processing component, to support a parallel model. This includes data acquisition and transfer.

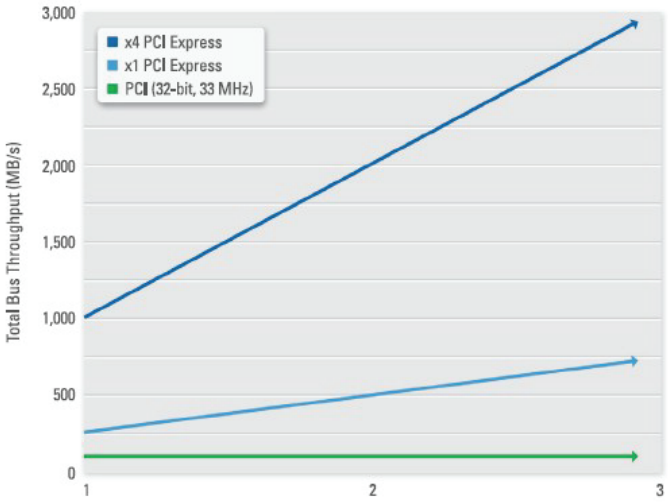

Figure 2. PCI Express delivers dedicated bandwidth instead of shared data transfer, significantly increasing the amount of data that you can acquire and stream to disk.

The most common data transfer buses for modular instrumentation today – including PCI, USB, Ethernet, and GPIB – do not support a truly parallel data transfer model because the devices on the bus share bandwidth. If the cumulative rate of acquisition or generation of the I/O devices is faster than the rate at which the bus is available, data could be lost. A common

solution to this problem is performing measurements sequentially and integrating large buffers of onboard memory on the I/O device so data is not lost while waiting for availability on the communications bus.

In contrast, PCI Express, the latest in high-performance data transfer buses, delivers dedicated bandwidth for each device while still providing throughput higher than any of the other commercial communications buses previously mentioned. Available in x1, x4, x8, and x16 lanes (pronounced "by 1," "by 4," and so on), PCI Express provides 250 MB/s of usable throughput per lane. The x1 and x4 options are most common for instrument-class hardware and provide 250 MB/s and 1 GB/s (four lanes at 250 MB/s) of dedicated throughput, respectively.

Figure 3. PCI Express and modular instrumentation provide a scalable model for parallel measurements.

As a PCI Express board acquires data, it is transferred from onboard memory across a dedicated PCI Express lane and streamed to a hard drive or into system memory. Once in system memory, the LabVIEW application can access the data, and if used with a multicore processor, the parallel measurement, from signal to final measurement, is completed.

Parallel Test

With multicore processing, PCI Express, and LabVIEW, test architects can create parallel measurement systems capable of testing a single unit under test (UUT) at a time. The definition of parallel test, however, is multiple UUTs undergoing testing simultaneously. The alternative is to test UUTs sequentially, one after the other. Although parallel test clearly reduces aggregate test times, increases test throughput, and improves instrument usage (see Figure 4), the complexity of developing a parallel test system can be prohibitive. Developing test management software that implements the testing of multiple UUTs at once requires a low-level understanding of how the operating system works with parallel operations, such as Windows Critical Sections, and careful consideration of how to implement instrument sharing among many UUTs without creating conflicts, or deadlocks.

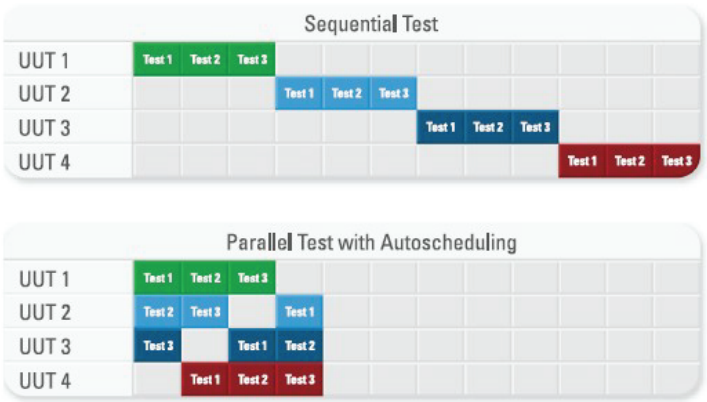

Figure 4. Testing multiple UUTs in parallel reduces aggregate test times while improving instrument usage.

An alternative to developing a custom parallel test system from scratch is to use off-the-shelf test management software, such as NI TestStand. NI TestStand abstracts the low-level complexity of parallel test system development using built-in features for executing parallel test sequences in multiple threads and managing both operating system and instrument resources.

Completing the Parallel Test Architecture

Multicore processors and PCI Express are changing the landscape of the modern PC and are helping LabVIEW graphical programming deliver on the promise of truly parallel processing and parallel measurements based on the data flow of an engineer's application logic. When NI TestStand is also employed, the parallel test architecture is complete. The results are test systems with increased data throughput, using PCI Express; increased processing power, using LabVIEW and multicore processing; and ultimately decreased total test cost per UUT, using NI TestStand.