Testing Multiple UUT’s at Once on Functional Test Systems

Application Note 1470

Tom Fay, Software Architect Agilent Technologies

Introduction

Manufacturing test engineers often face the challenge of testing multiple UUT’s (Units Under Test) on a single functional test system. Testing a single UUT functionally is hard enough, and testing more than one UUT on a single test system simply adds to the problem. Even so, there are often good reasons to test multiple UUT’s at once. Sometimes there is a requirement to do so: for example, modern PCB manufacturing often results in multiple boards physically built on the same panel for reasons of cost. In this case, the test system needs to be able to deal with multiple UUT’s.

Testing multiple UUT’s at once can also frequently help reduce the cost of test equipment required to match a particular manufacturing capacity by increasing test throughput per UUT and increasing test system utilization.

When testing multiple UUT’s together is desired, there are three main approaches to test:

• Sequentially test each UUT. (fully test UUT1, then UUT2, ...)
• Synchronous interleaved test of each UUT (Test1 on UUT1, Test1 on UUT2, ... Test N on UUT1, Test N on UUT2...)
• Full asynchronous test of each UUT (Test each UUT as a separate asynchronous process or thread)

Each approach has advantages and disadvantages, and each is suitable for specific types of applications.

Economics of Multiple UUT Testing

The choice to test multiple UUT’s at once is often an economic decision. Regardless of the approach, testing multiple UUT’s on a single test system will be more complicated than testing one UUT at a time. This can result in higher test development costs, but frequently this offset by the potential for reduced overall test costs. Reduced test costs are possible because the total time to test multiple UUT’s on a single system at once is often faster that the sum of the time to test each UUT one after another on that same system. This potential time savings can reduce the number of test systems needed for a production, resulting in a test cost savings. It is this cost savings that often motivate multiple UUT testing approaches.

On a manufacturing line, the testing process step must keep up with the rate of the other process steps. Otherwise the test process becomes a bottle-neck reducing the production capacity of the line. So if test is
too slow, the simplest approach is to duplicate the test systems so that multiple UUTs can be tested in parallel. With enough test systems, nearly any process step time can be met, but the cost may be prohibitive. To reduce the number of test systems and this cost, a natural idea is to test multiple UUT’s on each system. To determine how great the cost savings might be, there are several questions to answer: How long does it take to test N UUT’s at once compared to testing a single UUT, and how much does each multiple UUT test system cost compared to a single UUT test system?

If the line process step rate needed is P seconds, and the test system can test a UUT in T seconds, then the line needs (T/P) test systems rounded to the next whole number of systems to meet the line rate. If each test system costs C, then the total solution cost is:

\[ C_{\text{baseline}} = C \times \left( \frac{T}{P} \right) \]

Suppose each multiple UUT test system can test N UUT’s in a time T*TR seconds, where TR is the ratio of the multiple UUT test time to the single UUT test time. The effective test time per UUT is then (T*TR)/N. So \( \left( \frac{T\times TR}{N} \right) / P \) test systems will be needed to meet the process step rate. If each such test system costs C*CR, where CR is the ratio of the multiple UUT test system cost to the single UUT test system cost, the total solution cost is then:

\[ C_{\text{multiple}} = C*CR \times \left( \frac{(T\times TR)}{N} / P \right) \]

or

\[ C_{\text{multiple}} = \left( \frac{C*TR}{N} \right) \times C_{\text{baseline}} \]

Thus the \( \left( \frac{C*TR}{N} \right) \) overall cost ratio gives a measure of how much can be saved with a multiple UUT test solution. In general, the cost ratio and test time ratios cannot be less than one and often are greater. However, if the number of UUTs tested by each system is sufficiently large, the resulting test solution can often be much cheaper.

The development cost for a test solution is another cost worth analyzing. It stands to reason that developing a multiple UUT test system will be more difficult than one for a single UUT. The test system itself often requires more instrumentation and switching, and the fixturing to connect to the UUT’s is more complex. Similarly, the test system software is often more complicated. The test program for multiple UUT’s often is larger simply because of the number of UUT’s. Further, complicating factors like the need to keep separate test results for each UUT, the need to avoid testing empty UUT positions, and similar considerations can quickly make multiple UUT test software much more difficult.

Given this economic analysis, there are three attributes that are important to analyze for each multiple UUT approach:

- **The cost of the test system:** costs involved include the cost of the test solution hardware as well as the cost of the floor space occupied by the test solution. For the purposes of this paper, cost of maintaining the test solution will be ignored, but test engineers do need to think about this issue too.
• The development cost to create the test solution: development includes work to build the hardware portion of the solution, develop the software portion, and integrate both together.

• The speed of test: testing multiple UUT’s on a single test system will almost always be slower than testing each UUT on a dedicated test system in parallel, but it can often be faster than testing each UUT in series on one test system.

Armed with this framework, several different approaches to testing multiple UUT’s can be analyzed.

**Sequential Testing**
When faced with the challenge of testing multiple UUT’s at once, the first thought is often to simply test each UUT one after another. This is one step away from each UUT coming down an assembly line and being tested on the test system as it passes by: the difference is that all UUT’s are attached somehow to the test system at the same time.

This scheme is the most straightforward, and it does allow using a single test system to do the testing. It does often require an extra layer of switching or some mechanical solution to connect to the UUT to be tested next. This may be as simple as an operator switching connections to each UUT in turn and running the test, or as complicated as some robotic or mechanical solution that automates connecting to each UUT in turn. Either of these solutions injects additional time between the testing of each UUT to make the connection change, and this time can become significant. If an extra layer of switching is used, this time can be reduced.

**Match against the test attributes:**

• The cost of the test system: this scheme does use a single test system, but may have extra mechanical costs.

• The development cost: this scheme does minimize development costs since the test software is very similar to the software needed to test a single UUT.

• Test speed: this scheme falls short here, since the test time will never be better than the sum of the times required to test each UUT separately. The only time savings may be the simultaneous load and unload of all UUT’s. This scheme saves time only if load/unload operations are a significant part of the test time.
Application Considerations:

• Where the time of test is not important, like burn-in or durability test, this test strategy may be the simplest one.

• For test in manufacturing, this approach likely will be attractive only if load and unload times are relatively long or the process step time is relatively long for other reasons.

Synchronous Interleaved Testing

A more complicated approach to testing multiple UUT's is to interleave the test of the modules together in a structured way by doing particular tests for each UUT before progressing to the next test.

Using this approach, it is often possible to shorten the overall test time compared to the fully sequential approach outlined above. As the above diagram illustrates, simply performing the tests in this interleaved order can speed testing. This is because testing frequently requires programming instruments to particular settings, but once those settings have been made, multiple tests of the same type go faster. The synchronous interleaved testing method can take good advantage of this fact: it is always repeating the same test multiple times, once for each UUT. There are other characteristics of test like testing delays required to allow instruments and the UUT to reach the right states for test that can also be exploited to speed testing.

This scheme requires switching controlled by the test system to accomplish the test interleaving without incurring large mechanical handling delays. Further, the test system often has to keep track of the test results for each UUT separately even though testing is done all together. This can make test development more complicated without tools built into the test system to aid this development effort. In fact, some modern test executives do provide tools to allow a test program little more complicated than the single UUT case to test multiple UUT's using this approach.

Match against the test attributes:

• Test system cost
  a single test system is used with no requirement to duplicate testing resources other than switching. For many systems with excess switching capacity to begin with, this results in a test system cost the same as the single UUT case.

• Development cost:
  costs need not be significantly greater than the single UUT case with the right system software.

• Test time:
  Test time is often much less than the time for the previous sequential case. Time savings come from operations done once for all UUT’s, better repeated test instrument usage optimization, and shared set-up delays.
Application Considerations:

• Best when system software eases the development effort.

• Best when test applications do not require dedicating expensive instruments to a single UUT for long periods of time.

• Best when test systems have adequate switching capacity and control to perform the inter-UUT switching.

Automotive electronic UUT testing is an example where this approach can be attractive, particularly for smaller UUT’s. For example, on the Agilent TS-5400 automotive test system, the Agilent TestExec SL system software lets test engineers easily convert a single UUT test program into one for multiple UUT’s. The resulting test program is little more complicated than the single UUT case, since Agilent TestExec SL can take care of the inter-UUT switching and test interleaving automatically. In one real-life case, a customer was able to test 6 UUT’s in 1.5 times the time to test one on a system with no extra cost over the single UUT system cost. The resulting cost ratio from the economic model is:

$\frac{(CR \cdot TR)}{N} = \frac{(1.5 \cdot 1)}{6} = _$

So multiple UUT test reduced the cost of test by a factor of 4 for this UUT.

Full Asynchronous Test

Even PC’s now support multiprocessing where multiple processes can be interleaved automatically to use the wait times for one process to do work for another process. Why not take advantage of this capability for testing multiple UUT’s? The general idea is to delegate the testing of each UUT to an independent process or thread, then let the operating system manage overlapping of testing between these processes. The start of testing for any particular UUT can be fully independent from the start of test for other UUTs, which can be attractive if the UUTs don’t arrive at the test system at the same time. Further, as testing completes for a UUT, it can be unloaded and a new one loaded in its place while testing continues on the other UUT positions. This helps if the time to test UUT’s varies, since it avoids having UUT positions sit idle while testing completes on other positions.

Each testing process focuses on testing one UUT, so the test program for each process ought to look very similar to the single UUT testing case. However, multiprocessing test has complications that can make this approach hard to implement.
The primary problem is sharing of resources like instruments or I/O channels between test processes. Sharing instruments requires a test process to lock instruments while it is using them since the instrument might not stay in the state the test process needs long enough otherwise. An instrument locked by one test process cannot be used by any other test. So instruments shared between UUTs should be locked only for short periods to avoid testing delays. If each UUT has dedicated instruments, this is not an issue, but locking the I/O channel (GPIB or VXI interface, for example) for long periods can still cause problems. As a result, the drivers for all instruments need to minimize instrument-wait locking and I/O channel locking times: that is, they need to be nonblocking. Since this nonblocking behavior is not required for the single UUT test case, standard drivers often block on instrument queries to simplify driver writing, and these drivers can cause problems for this multiple UUT test scheme.

Deadlock is another complication if many instruments will be shared between testing processes. Deadlock is the classic computer science problem where one testing process is using instrument A and needs instrument B to complete its measurement while another testing process is using instrument B and needs instrument A to complete its measurement. In this case, both testing processes are stuck in a deadlock. Neither can finish its work and release the instrument needed by the other. This is not a good thing to happen on a manufacturing line, to say the least. The order in which a test program uses instruments can be managed to avoid this problem, however.

Multithreading can cause an additional problem for this style of multiple UUT testing. Multithreading is a low-overhead way of supporting parallel processing in a single application process. Multithreading does this by having all threads share the same process address space. As a result, all software in the application must be carefully designed to allow multiple threads to use the same code and memory without that code getting confused about which thread is doing what. Again, instrument drivers written for single UUT testing often don’t have the careful programming needed to support multithreading. In contrast, if multiple UUT testing is accomplished using multiple processes operating in separate address spaces, no such problems arise even with drivers designed for single UUT testing.

**Match against the test attributes:**

- **System cost:**
  test systems are cheaper to the extent they share hardware and infrastructure between UUTs, especially when the test system can still fit in a single rack to save floor space.

- **Development cost:**
  test solution development looks very similar to the single UUT test development if instrument drivers are available to facilitate concurrent test.

- **Test time:**
  testing on all modules can be nearly as fast as testing one module. This scheme can handle UUT's arriving at the test system at different times and can immediately load another UUT to test as soon as any of the UUT’s being tested fail.
**Application Considerations:**

- Best when testing is inherently slow with significant wait-times.
- Best when testing requiring a dedicated instrument for long periods can be handled with replicated instruments that do not require constant supervision by the test system controller.
- Best when nonblocking instrument drivers are available.

Cellular telephone testing is an example which matches these considerations. For example, the Agilent TS-5500 family of cellular phone test systems, also based on Agilent TestExec SL, use this approach to multiple UUT testing. Special concurrency-enabled drivers for the instruments on those test systems assure that no one test process blocks other test processes for long periods, and each UUT has a dedicated RF test set managing the timeconsuming RF testing for the cellular phone. A single operator interface manages a separate Agilent TestExec SL process for each UUT to avoid threading problems, and the operating system is left to manage the test operation interleaving. Since the same test program is used for each process, avoiding deadlock problems is much easier. In this case, the cost advantage is shared infrastructure hardware for the controller, VXI or PC plug-in instruments, and rack coupled with simplified test development. In addition, because the components are treated as part of a system, they can be optimized to work together faster in a way that a custom-built test system might not. The Agilent TS-5530 Cellular Phone Functional Test System

**Summary**
Three different methods of testing multiple UUTs on a single test system were explored:

- **Sequential Test:**
  Good for simple cases where test speed is not important.

- **Interleaved Synchronous Test:**
  Good for minimizing the test system hardware and speeding up test when instruments are used in short bursts.

- **Asynchronous Test:**
  Good for minimizing test system costs and speeding up test when instruments must be dedicated to a particular UUT for long periods.

The first method is the simplest, but may not provide enough benefit to warrant using it unless multiple UUT’s must be tested together for some other reason. The second method is attractive when system software supports it and the application allows it. The third method is attractive if instruments must be dedicated to a particular UUT anyway and nonblocking drivers enabling concurrency are available.
Agilent Technologies' Test and Measurement Support, Services, and Assistance

Agilent Technologies aims to maximize the value you receive, while minimizing your risk and problems. We strive to ensure that you get the test and measurement capabilities you paid for and obtain the support you need. Our extensive support resources and services can help you choose the right Agilent products for your applications and apply them successfully. Every instrument and system we sell has a global warranty. Support is available for at least five years beyond the production life of the product. Two concepts underlie Agilent’s overall support policy: “Our Promise” and “Your Advantage.”

Our Promise

Our Promise means your Agilent test and measurement equipment will meet its advertised performance and functionality. When you are choosing new equipment, we will help you with product information, including realistic performance specifications and practical recommendations from experienced test engineers. When you use Agilent equipment, we can verify that it works properly, help with product operation, and provide basic measurement assistance for the use of specified capabilities, while your instrument is under warranty or technical support contract. Many self-help tools are available on Agilent’s web site.

Your Advantage

Your Advantage means that Agilent offers a wide range of additional expert test and measurement services, which you can purchase according to your unique technical and business needs. Solve problems efficiently and gain a competitive edge by contracting with us for calibration, extra-cost upgrades, out-of-warranty repairs, and on-site education and training, as well as design, system integration, project management, and other professional engineering services. Experienced Agilent engineers and technicians worldwide can help you maximize your productivity, optimize the return on investment of your Agilent instruments and systems, and obtain dependable measurement accuracy for the life of those products.

Agilent Email Updates

www.agilent.com/find/emailupdates
Get the latest information on the products and applications you select.

Agilent T&M Software and Connectivity

Agilent’s Test and Measurement software and connectivity products, solutions and developer network allows you to take time out of connecting your instruments to your computer with tools based on PC standards, so you can focus on your tasks, not on your connections. Visit www.agilent.com/find/connectivity for more information.

By internet, phone, or fax, get assistance with all your test & measurement needs

Online assistance:
www.agilent.com/find/assist

Phone or Fax

United States:
(tel) 800 452 4844

Canada:
(tel) 877 894 4414
(fax) 905 282 6495

China:
(tel) 800 810 0189
(fax) 0800 650 0121

Europe:
(tel) (31 20) 547 2323
(fax) (31 20) 547 2390

Japan:
(tel) (81) 426 56 7832
(fax) (81) 426 56 7840

Korea:
(tel) (82 2) 2004 5004
(fax) (82 2) 2004 5115

Latin America:
(tel) 305 269 7500
(fax) 305 269 7599

Taiwan:
(tel) 080 004 7866
(fax) (886 2) 2545 6723

Other Asia Pacific Countries:
(tel) (65) 375 8100
(fax) (65) 838 0252
Email: tm_asia@agilent.com

Product specifications and descriptions in this document subject to change without notice.
© Agilent Technologies, Inc. 2004
Printed in the USA February 11, 2004
5989-0670EN