

Implementation of Wafer Level Parallel Test

Randall Lee
Keithley Instruments, Inc.

PARALLEL parametric test is an emerging strategy for wafer-level testing that involves concurrent execution of multiple tests on multiple scribe line test structures. It offers a relatively inexpensive way to increase throughput, thereby lowering the cost of ownership (COO) significantly. Just as important, as device scaling increases the randomness of failures, parallel testing can address the growing need to perform more tests on the same structures in less time. In this case, users can choose to either increase the number of tests performed at each site, or increase the number of sites.

Making the transition from strictly sequential parametric test to the use of parallel test techniques can appear daunting, even to experienced parametric test engineers. The best way to approach this challenge is to break down the process into a number of smaller, more attainable phases. (See *Figure 1*.) Parallel test doesn't necessarily demand test structure modifications or developing new structures for new processes—there's plenty of potential for reducing test times or

increasing the number of measured parameters even when continuing to test existing structures.

Planning For Parallel Test

Parallel Test Candidates – Parallel test is appropriate for virtually any solid-state technology. It's just as suitable for gallium arsenide processes as it is for mainstream silicon processes. There are only a few minor caveats associated with selecting a process for parallel test:

1. The test structure shouldn't introduce instability into the measurement by being tested in parallel with another structure. Structures with shared terminals at any level, whether in the diffusion or interconnect layer, have the potential to produce skewed results. Unfortunately, these shared terminals are fairly common in legacy structures. Test structure designers often use common pads for multiple devices under test (DUTs) to conserve space in crowded scribe lines. More information on parallel testing of existing scribe line Test Element Groups

(TEGs) will be covered in a subsequent article.

2. Particularly for new device technologies, it's essential to establish a measurement baseline using sequential testing prior to implementing parallel test. Variations in device performance are more common with new technologies than with existing ones. Given that one of the objectives of parametric testing is to understand where the variations in the process are and then to reduce them through the development process, it's critical to establish this sequential test baseline; parallel testing may introduce additional variations as a result of either tester timing or device interference. Without a sequential test baseline for comparison, it's impossible to distinguish between "new device" variations and "parallel test" variations. When using a Keithley parametric test systems the company's *pt_execute* software provides a toolset and coding method for parallel test that allows switching from sequential to parallel testing quickly and easily. It also manages test resource allocation.
3. For new processes, the best time to turn parallel test "on" is at the beginning of a volume ramp. This strategy offers the greatest bang for the investment buck by reducing the number of testers needed as the product goes into volume production. **Note:** It is best to learn how to use parallel test on a mature process, not during ramp-up of a new process.

Identify Prober Throughput Limitations

– Before test engineers begin designing new structures or modifying test sequences to implement parallel test, it's critical that they consider everything that affects the throughput of the test cell as a whole, not just raw tester speed. Weighing the impact of any prober throughput limitations is an important first step in ensuring the implementation effort achieves the maximum potential test time reduction. The primary timing parameters affecting prober throughput are:

- First wafer load and align times (typically ~90 seconds)
- Site index time (typically ~600–700ms)
- Subsite index time (typically ~350ms)
- Wafer swap time (typically 45 seconds)
- Last wafer unload (expel) time (typically ~30 seconds)

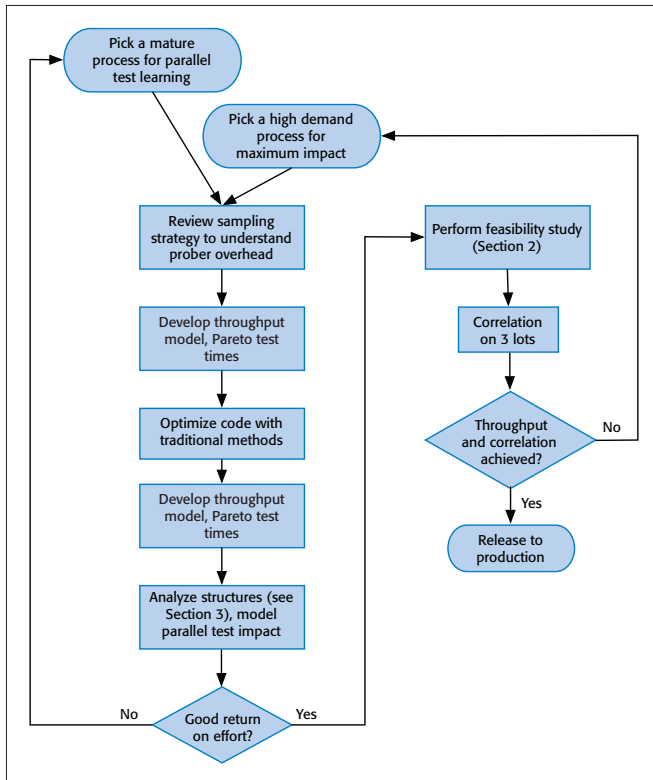


Figure 1. Basic parallel test implementation strategy.

The typical times listed above do not reflect any specific prober's performance and are offered simply to provide an indication of magnitude. The exact times for each parameter will depend on the mechanical design trade-offs of a particular prober. When designing test structure sets for parallel test, typically one of the goals is to minimize the number of subsite moves in order to reduce the impact of subsite index time on throughput. However, all the prober throughput parameters must be considered and factored into the test cell's overall throughput budget and return on parallel test implementation effort.

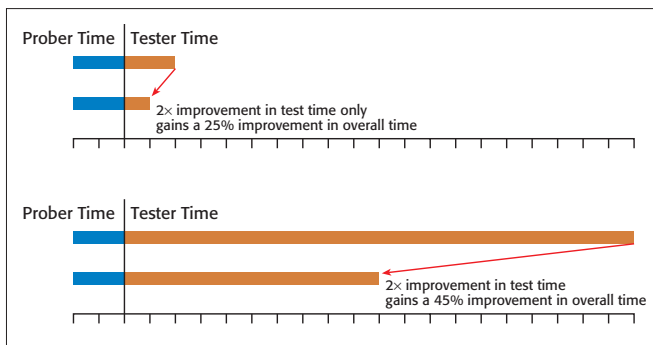


Figure 2. Prober overhead can dilute the gains of parallel test.

The potential gains from parallel test can be greatly diluted by prober overhead. The closer the ratio of tester time to prober time, the more that dilution occurs. In the examples in *Figure 2*, when tester time is equivalent to prober time, it is possible to produce a 2x improvement in throughput, but only a 25% overall improvement. With a larger tester/prober time ratio, an overall gain of 45% is possible. Still, on a heavily loaded test floor, a 25% throughput gain may

provide the needed ROI to justify the implementation effort.

Evaluate Existing Sampling Strategy – Over the manufacturing lifespan of a device, the number of sites and subsites tested on each wafer will typically undergo substantial changes. Fabs generally test the most in the early weeks after the device enters production and then reduce this regimen to a smaller set of tests on far fewer sites and subsites as the process reaches maturity. Those test sequences and site/subsite numbers will also depend significantly on who's doing the testing and their business objectives. When fewer tests are performed on fewer sites and subsites—thereby reducing the amount of time the prober requires for site and subsite indexing—the other prober timing parameters (first wafer load and align times, swap times, and unload times) assume greater significance in the overall throughput picture.

Keithley recommends implementing parallel test for the first time on a mature existing process, rather than on a new product with new test structures. The knowledge gained during implementation on a well-known process will provide valuable insights for subsequent implementations on newer products. In addition, the necessity of ramping up production on a new product as quickly as possible also makes it unlikely that a fab would spare the tester capacity and human resources necessary for a throughput improvement project. In fact, parametric test experts recommend employing conventional throughput improvement techniques first, given that these approaches offer more straightforward throughput benefits, even when parallel test techniques are impractical.

Prior to attempting to implement parallel test on an existing process, a team of test engineers must perform an in-depth feasibility study. This study, which starts with a review of the documentation for the wafer's existing test structures, allows the implementation team to determine the most appropriate DUT test groupings to maximize the reduction in test time. (A test grouping is the set of sequential tests that will be performed in parallel.) This job falls to the test engineers because they have the best understanding of the test resources (number of SMUs and other sourcing and measurement instrumentation) available on specific test systems. They also have the most insight into the structures themselves and which ones can be grouped, in addition to the greatest understanding of the tests currently being performed and those likely to continue being used as the process approaches maturity.

Evaluate TEGs – As part of the feasibility study, the implementation team must evaluate the opportunities for test time reductions that the TEG offers when pairing “like” tests, such as multiple I_{ON} or V_T tests, keeping in mind that pairing low-level measurements may increase the variability of the test results. Pairing long-duration tests that are performed on the same type of structure is another possibility for test time reduction. One example is pairing two gate oxide integrity (GOI) tests performed on two different gate dielectrics within the same TEG. It's also important to evaluate the various test grouping options, based on the tester resources available (i.e., the SMUs and other instruments).

In actual parallel testing, this grouping function may be performed automatically by parametric tester software, such as Keithley's *pt_execute*. However, it's important to gain an early awareness of the level of throughput improvement possible with the existing tester

configuration. For example, if the test engineer puts three V_T test in order, these tests are performed in that order.

When *pt_execute* is used, the test grouping process is based on the resources available (pins and instruments) and limitations imposed by test conditions that must be executed separately. For example, if the test engineer has four V_T tests paired together and each V_T test employs three SMUs, *pt_execute* can only perform two V_T tests at a time (in parallel). Therefore, only two V_T tests are grouped together. If the test grouping review indicates the current hardware configuration offers limited throughput improvement, it may be a sign that the fab should consider investing in additional SMUs for the tester in question.

Evaluate and Create Test Program Algorithms – Long duration tests that represent fundamental elements of the manufacturing process, such as the GOI test mentioned previously, are the best candidates for both the feasibility study and performing tests in parallel. During the feasibility study, the team can prevent wasted time by not bothering to evaluate tests that are likely to be eliminated as part of the typical test time reduction activities, (i.e., the usual test streamlining that goes along with increasing process maturity.) It would be unwise to base a parallel test cost justification decision, or rewrite sequences on tests that are likely to be deleted, unless the ROI for the time such structures are used is very compelling. This could be the case during technology development or ramp-up. One example of this that might deserve special consideration is testing used for structure-related debug, such as tests on structures intended to monitor silicide formation. Another example is process-maturity-related debug, such as tests on comb or serpentine structures used to monitor yield variability.

The next step in the feasibility study is to create a set of test algorithms for a conservative DUT test grouping. In this instance, “conservative” means a sequence that includes no low-level tests, such as leakage tests (because of their potential for disruption by other tests, such as breakdown tests), or non-alike groupings (such as attempting to test breakdown voltage and voltage threshold at the same time). Once the algorithms for this set of tests are complete, they should be

loaded and run on the tester in both sequential and parallel modes. Then, data from the two different test modes should be compared carefully.

These sequences can be run and the data can be taken at the sub-program level. In other words, rather than testing an entire wafer, complete with prober indexing from one subsite to the next, this comparison process requires only looping through the tests for a single subsite. The comparison also involves close examination of the control charts for both the parallel test run and the sequential test run, which allows the implementation team to identify unintended offsets or interferences that parallel testing may introduce. Even though this exercise provides only a comparison of the sequential vs. parallel test execution times, it offers an important indication of the potential for overall test time reduction. Again, *pt_execute* software facilitates switching parallel testing on and off to quickly gauge the impact on test execution time of any code changes, and helps track down the source of correlation problems.

Parallel Test Development Process

Select Algorithms for Re-use and Modification – Once the feasibility study is complete, it’s time to review which of the existing test algorithms (or macros) can be modified and reused in a parallel test environment (typically, roughly two-thirds of them). Another benefit of the *pt_execute* package is that it helps increase the percentage of existing test libraries that can be reused in parallel test and facilitates the creation of new algorithms. However, it’s important to keep in mind that a macro-by-macro review process is very beneficial. Typically, this leads to some rework of almost all algorithms as the implementation team recognizes reductions in test times that are possible through adjustments to delays and integration times.

Create New Macros – In one sense, creating new macros involves a “deconstruction” process for some implementation teams. For example, the test engineer who created the original sequential test program may have been especially diligent about reducing tester overhead. He or she may have grouped the “connect” statements for all the instruments applied to a specific set of test structures on a subsite, and then set up force/measure sequentially on all of the structures.

This practice of grouping many tests into one large sequence is often referred to as writing jumbo algorithms.

Unfortunately the earlier use of these “pseudo-parallel test” sequences typically forces the implementation team to take a step back and, in effect, start over. To gain the throughput benefits of parallel test, they need to develop far simpler, single-purpose algorithms designed to be performed in parallel with other single-purpose algorithms on a single DUT. (Adding *pt_execute* commands at the appropriate points in the test sequences will automatically associate the macros with the appropriate DUTs.)

Conduct Correlation Studies – Correlation studies done initially at the sub-program level must become an ongoing process. It is critical to compare test results obtained in sequential and parallel modes throughout the development process and eventually at the composite program level. While it’s obviously satisfying to identify test execution time reductions of up to 60% at the sub-program level on a single subsite, it’s much more important to see significant throughput gains on the composite program level, which also includes the subsite and site indexing times.

Keithley typically recommends analyzing and correlating the data from three wafer lots for gauge performance and throughput modeling. This stage can reveal new test issues, such as probe card charging or other problems. These must be resolved before the implementation process can be considered complete.

Ongoing Implementation

As is true with virtually any type of implementation process, implementing parallel test tends to be somewhat easier the second time around. This is especially true when the original implementation team is diligent about documenting their efforts and sharing that knowledge with colleagues through a formal “Best Known Method” process.

Subsequent parallel test implementations on new wafer designs may allow a significantly larger throughput reduction than was possible on legacy test structures. The lessons learned in the first implementation can lead to the creation of new test structures optimized for parallel test. For example, a number of device manufacturers with


experience in parallel test choose to incorporate the devices associated with all their long duration tests into one test structure. Others take advantage of the flexibility that parallel test's higher test execution speed offers to add new tests or more test devices, so they can gather levels of information that were previously impractical.

In an ideal testing world, every test structure would be very simple, totally isolated electrically, and equipped with a pad for every DUT terminal. Oddly enough, this is somewhat similar to the test structure design philosophy typically followed during technology development, when the objective is to obtain the highest possible data granularity. This is achieved by testing many of the same

types of devices with various gate lengths and structures with contact chains of various lengths, etc.

Conclusions

Obviously, every team will have its own timetable for implementing parallel test. The time required depends on organizational priorities and resources available, including test cell capacity, test engineers, and structure designers. However, in Keithley's experience with fabs that are successfully using its S680 tester system, they should generally plan on a first implementation of parallel test to take approximately three months, from the implementation team's feasibility study to the final switch over.

More information on wafer level parallel parametric test can be found in Keithley's Parallel Test Technology handbook, available at www.keithley.com/at/508. 

About the Author

Randall Lee is a Senior Industry Market Manager in Keithley Instruments' Semiconductor Test Group where he is responsible for parametric tester product marketing and management. He received a BS in Electrical Engineering from Rensselaer Polytechnic Institute, and has 25 years of experience in semiconductor lithography and wafer characterization and test.

Specifications are subject to change without notice.
All Keithley trademarks and trade names are the property of Keithley Instruments, Inc.
All other trademarks and trade names are the property of their respective companies.

KEITHLEY

A G R E A T E R M E A S U R E O F C O N F I D E N C E

KEITHLEY INSTRUMENTS, INC. ■ 28775 AURORA ROAD ■ CLEVELAND, OHIO 44139-1891 ■ 440-248-0400 ■ Fax: 440-248-6168 ■ 1-888-KEITHLEY ■ www.keithley.com

BELGIUM

Sint-Pieters-Leeuw
Ph: 02-3630040
Fax: 02-3630064
info@keithley.nl
www.keithley.nl

CHINA

Beijing
Ph: 8610-82255010
Fax: 8610-82255018
china@keithley.com
www.keithley.com.cn

FINLAND

Espoo
Ph: 09-88171661
Fax: 09-88171662
finland@keithley.com
www.keithley.com

FRANCE

Saint-Aubin
Ph: 01-64532020
Fax: 01-60117726
info@keithley.fr
www.keithley.fr

GERMANY

Germering
Ph: 089-84930740
Fax: 089-84930734
info@keithley.de
www.keithley.de

INDIA

Bangalore
Ph: 080-26771071, -72, -73
Fax: 080-26771076
support_india@keithley.com
www.keithley.com

ITALY

Peschiera Borromeo (Mi)
Ph: 02-5538421
Fax: 02-55384228
info@keithley.it
www.keithley.it

JAPAN

Tokyo
Ph: 81-3-5733-7555
Fax: 81-3-5733-7556
info.jp@keithley.com
www.keithley.jp

KOREA

Seoul
Ph: 82-2-574-7778
Fax: 82-2-574-7838
keithley@keithley.co.kr
www.keithley.co.kr

MALAYSIA

Penang
Ph: 60-4-656-2592
Fax: 60-4-656-3794
chan_patrick@keithley.com
www.keithley.com

NETHERLANDS

Gorinchem
Ph: 0183-635333
Fax: 0183-630821
info@keithley.nl
www.keithley.nl

SINGAPORE

Singapore
Ph: 65-6747-9077
Fax: 65-6747-2991
koh_william@keithley.com
www.keithley.com.sg

SWEDEN

Solna
Ph: 08-50904600
Fax: 08-6552610
sweden@keithley.com
www.keithley.com

SWITZERLAND

Zürich
Ph: 044-8219444
Fax: 044-8203081
info@keithley.ch
www.keithley.ch

TAIWAN

Hsinchu
Ph: 886-3-572-9077
Fax: 886-3-572-9031
info.kei@keithley.com.tw
www.keithley.com.tw

UNITED KINGDOM

Theale
Ph: 0118-9297500
Fax: 0118-9297519
info@keithley.co.uk
www.keithley.co.uk