

Testing the Tester: What Broke? Where? When? Why?

Alfred L. Crouch
Inovys Corporation
Austin, TX

In recent times in the business of semiconductor testing, the claim has been that Test Cost has begun to dominate the overall cost of product (die + packaging + test) in some product lines. The cost models that I have seen are simplistic models that just take into account the product yield and the per-second test cost. More recently, I have become even more concerned over the effects of Mean-Time-Between-Failures (MTBF) and Mean-Time-to-Repair (MTTR), and have found that they are also quite significant in this cost equation (for wafers with thousands of die, downtime can cost in the tens of millions of dollars in throughput loss even for reasonable MTBF and MTTR).

Regardless of how the device is tested (Scan, BIST, Functional, etc.), the tester must segregate the various devices into various Pass and Fail Bins. The tighter the criteria, the more accurate the tester must remain during this process. For example, high frequency device operation with a tight operating range requires highly accurate edge placement and a very stable clock from the tester; or extremely low power consumption by the device requires a highly accurate current measurement capability that can exclude the tester's contribution to power consumption. If a device is logged as a Fail, there must be some reasonable expectation that the fail is due to the device, not the tester – a degraded tester can contribute to yield loss and may require the re-testing of parts. A degraded tester may also allow bad devices to be logged as Passed, and therefore, it may also contribute to test escapes – which may be caught in a more expensive test process at a later time (at least you hope it will be caught before it is in your customer's application). In addition, the ability to conduct multi-site parallel device testing may also require the calibration and diagnostic cycles to resolve to some small portion of the tester, as opposed to the whole tester.

All of this states that it is not just good enough to test the tester when it is manufactured, but that the tester must be continually checked for adherence to its specifications – even while devices

are being tested. The sooner degradation is found, the sooner the tester can be repaired or calibrated, and the fewer parts need to be retested. In the modern ATE world, this can be a problem at two extreme points – the high performance ATE, and the low cost DFT ATE. The high performance ATE is a problem because its requirements are so extreme: highly precise clocks, high bandwidth data rate; highly precise current control, etc. – the machine that tests the 'latest and greatest' technology must remain one step ahead of that technology. The low cost structural or DFT ATE is a problem because of the sheer amount of supported resources: generally, very high pin count and deep-deep test memory availability. Both of these extremes may push calibration and diagnostic times to unacceptable time costs (if test times are bad for a device – imagine the test time for testing an ATE with resources capable of testing 16 of those devices at once).

My opinion on the matter is that the tester needs to incorporate the same 'Cost Savings' DFT techniques that the devices use. Memory can be tested with Memory BIST – and repair and remapping algorithms should be used to ensure the test program is held in pristine memory; digital drive pins can be tested for timing and logic value with path delay scan techniques; clocks can be tested for timing/period and stability/jitter with some of the same generally available PLL BIST techniques. To be cost effective, the DFT-based testing must be accomplished in part or in whole as often as possible – when resources are idle, or between wafer or product changes. In addition, on-line concurrent diagnostic tracking needs to be employed to identify when in a test process that degradation begins to occur. This means that more information needs to be brought through the test process and be available to the tester server so that concurrent correlations can pinpoint when binning for passes and fails shows a trend. It all boils down to making sure that good parts are truly good and bad parts are truly bad – no matter where they reside.