‘Can DFT Totally Delete Traditional Functional Testing?’

David M. Wu, Intel Corporation
MPG/ Texas Development Center

Panel Synopsis:
Design-for-testability techniques have been around for more than two decades to help reducing high-end ATE dependencies and manual test pattern generation efforts. However, as device speeds break the GHz boundary and I/O bus transactions grow into the GTs range, the shrinking process feature size and aggressive process parameters may introduce new defect types and failure mechanisms that significantly reduce the effectiveness of traditional DFT techniques, such as scan and BIST. Furthermore, issues of test data volume and power dissipation during scan/BIST become more critical as target IC size/complexity increases.

Functional testing, on the other hand, targeted at functional and system paths and applied test patterns at the system speed. Historical results indicted that functional testing is very effective in achieving low DPM. However, as transistor count approaches 100 millions and beyond, the test development effort becomes excessive and the demand of extremely high speed, highly accurate and therefore highly expensive testers are becoming unacceptable to most semiconductor companies.

This panel will discuss traditional and new DFT techniques and ask the audience: ‘How can or why can’t D.F.T totally Delete the traditional Functional Testing?’

Panel members will be ‘assigned’ to one of the following two positions in order to stimulate debate. Each position statement is solely used for the purpose of this panel; it may not represent actual opinion of the panelist or his employer.

Position 1: (Steve Pateras, Rob Aitken, David Wu)
"There is NO WAY functional testing can keep up, DFT and BIST are the only way to go"

Position 2: (Burnell West, Bill Bottom, Bulent Dervisoglu)
"There is NO WAY that DFT will totally Delete Functional Testing”

Have fun!

Moderator: Edward McCluskey, Stanford CRC
Panelists: Steve Pateras (LogicVision), Burnell West (Schlumberger), Rob Aitken (HP), Bulent Dervisoglu (Cadence Design), Bill Bottom (3rd Millennium Test), David M. Wu (Intel).
Moderator statement for panel on

'Can DFT Totally Eliminate the Traditional Functional Testing?'

Edward J. McCluskey
Center for Reliable Computing
Stanford University

The moderator should be neutral. Thus, it would be inappropriate for me to take a position on the substance of the panel. I remember once attending a panel with a somewhat similar title. I was confused by this panel until I cornered some of the panelists after the session and found out that some of their statements meant something different from my understanding. I thought that I might help out with the present panel by suggesting some definitions in the hope that the panelists will tell us which definitions they are using.

**Functional Tests**

This is a marvelous term that has lots of meanings. Let's try to list some of them:

1. Test in which the part is operated in the same fashion as in the application;
2. Test in which a sequence of valid input signals is applied to the device under test and the output response is compared to the correct response. Only the logic values represented by the signals (not the exact voltages or currents) are considered.
3. Test that detect failures by "verifying correct operation" rather than by verifying the absence of specific faults.

*Discussion:* In connection with chip test, the functional test terminology is used to distinguish from parametric tests, but does not rule out scan testing, for example. The term *Boolean test* is sometimes used for this type of test.

**Structural Tests**

1. Any test that is testing the structure of the device rather than its function;
2. Any test in which the device is restructured during testing;
3. Any test using patterns generated from fault models;
4. Any test that tests interconnect between modules (JTAG, bed of nails).

**System Test**

1. Any test that tests not only a component but tests the system in its entirety;
2. Test of a single device performed by running that device in a system.

**Design for Testability**

Everybody knows what this means so we don't have to define it.

1. A design technique in which some features of the design are included specifically to facilitate testing;
2. A design technique that ensures that the design can be restructured during testing.

*Discussion:* Design for testability is often used to describe a design procedure that inserts internal scan (LSSD) into the design.

It is hard to stop writing definitions, but maybe these are enough to fit in with the panel discussions. I do not necessarily agree with all the alternative definitions, maybe the audience will help correct or add to these definitions.