Abstract

Today's electronic designs have become prone to errors and defects due to the ever increasing complexity and compactions. This has resulted into imparting of much more importance to VLSI testing. Testing is mandatory and has to be performed on each manufactured product. Low cost and good defect coverage are the basic goals of testing, which are again determined by fault models, test volume and time. Time depends on how the tests are scheduled. Test scheduling has therefore become an important area of research. The work here is devoted to test scheduling of 3D SoCs taking into account the severe challenge it faces for its adoption i.e. the thermal management problem. 3D technology fulfils the demand of faster and compact design but there is a sharp rise in power density in such arrangement. Due to vertical stacking in 3D technology, there is a sharp rise in temperature especially for the layers far from heat sink. Consequently formation of hotspots may occur which may lead to device failure. Testing dissipates more power than the functional power because of the high switching activity that takes place during testing. All this requires thermal aware based test scheduling so that temperature does not rise above limits. The method presented here involves a thermal aware test scheduling for a 3D Soc built up using floorplan of benchmark circuit d695.
Test Scheduling of Stacked 3D SoCs with Thermal Aware Considerations

and few other examples. The modeling of 3D structure is done using resistors and explores conductance mode of heat transfer. The method has been compared with sequential test scheduling since no other work on similar lines is available to the best of knowledge of authors. The method shows a marked reduction in temperature rise and consequent elimination of hotspot formation.

References

ICs: Motivation, performance analysis and design implications. In 37th ACM design Automation Conference, pp 873-880.

Index Terms
Computer Science
Software Engineering

Keywords
Thermal awareness  hotspots  3D SoCs.