System Chip Test: Are We There Yet?

Prab Varma
Veritable Inc.

System chip, system on chip (SOC), system on silicon (SOS), there are many terms and many definitions that have arisen to describe the large chips that are emerging as we move further into the murky depths of the deep sub-micron era. Almost everyone has their own definition of system chip, which may explain why it is that almost everyone can claim to be designing system chips. However, as they begin to navigate the uncharted waters of system chip design, many are beginning to realize why SOS is an appropriate acronym.

A system chip is defined here as a complete system implemented on a single IC such that no peripheral chips are required. Thus, a system chip typically contains embedded core logic (such as microprocessor and DSP cores), embedded memory and analog components. A key enabling component of system chips is the re-use of existing design components sometimes called Intellectual Property (IP) cores or virtual components (VC). By this definition, even after having been talked about for the last twenty years, the system chip era is still just dawning.

Design re-use is narrowing the system chip design productivity gap but test and verification are now becoming the bottleneck. The key issues in testing core based system chips are how to optimally provide test access and test isolation to allow test methods and test vectors to be re-used for an embedded core, while ensuring that area, performance, test time and ‘at-speed’ test requirements are met. Despite all the talk about reducing time to market and having millions of spare transistors available, a primary differentiating feature of hard cores is how small their silicon foot-prints are and so adding area for test is still a significant concern.

Scan-based structural testing is becoming more widespread but many core providers still supply only functional vectors. For example, the leading hard core provider, ARM Ltd., provides functional tests for its cores and generally proposes the use of a test bus based technique to deliver these vectors; although a boundary scan technique is also supported, it is prohibitive in terms of test vector application time.

Standards and meta-standards groups such as the IEEE P1500 task force and the Virtual Socket Initiative Alliance (VSIA) are working to facilitate the ‘plug and play’ of virtual components. The VSIA test development working group is working on two deliverables: test data interchange standards and VC provider test guidelines, while the IEEE P1500 task forces are developing a core test architecture and a core test description language. Standards work is progressing well. However, the industry cannot wait for standards and will march forwards without them, with the EDA tool providers leading the charge.

EDA tools that automate most of the system chip test integration process are emerging and, as standards solidify, these tools will gain in maturity and provide automated support for test logic integration and test vector assembly. However, the system chip test planning process will still rely heavily on designer expertise.

While we are still in the early stages of developing a core test methodology that is widely accepted, two promising test re-use techniques are emerging: the test bus and test rail methods. Both approaches address the test bandwidth problem by allowing expandable (1 to n bit wide) test data access. The test bus technique is well suited to the wide range of existing core test techniques such as functional testing, scan and BIST, while the test rail is a natural extension of today’s boundary scan techniques and works well with structured test methodologies. Both techniques have strengths and it is likely that a mixture of both approaches will continue to be adopted.

BIST is a much heralded virtual component test technique and will continue to grow in importance but building an ATE on every chip is unlikely to become cost-effective in the foreseeable future. It is more likely that scan and functional test methods will continue to be the dominant virtual component test methods.