Implementing Design-for-Test within a Tile-Based Design Methodology - Challenges and Solutions

Venkat Yellapragada, Suresh Raman, Banadappa Shivaray
Xilinx
India

Ashok Anbalan
Mentor, A Siemens Business
India

Luc Romain, Benoit Nadeau-Dostie, JF Cote, Albert Au
Mentor, A Siemens Business
Canada

Giri Podichetty, Martin Keim
Mentor, A Siemens Business
USA

***Abstract*—A tile based design methodology consists of developing design blocks that are inserted in design layouts by placing blocks next to each other, making a tile-to-tile connection by abutting corresponding physical signal lines at the border of the tile. Very large systems can be easily and rapidly developed by seamlessly integrating tile elements in the layout. Further, the ease of top-level integration underlines the advantages over a bottom-up approach. However, this tile-based approach is incompatible with traditional DFT tools, which were created to work in accordance with the bottom-up design methodology. This paper outlines some of the obstacles to overcome, to support a truly tile-based DFT methodology. We describe here a working solution for a large production design, underlining a successful implementation of a tile-based Memory Test methodology.**

Keywords—tile-based design, bottom-up design, Memory BIST, Memory Repair, IJTAG, IEEE 1687

I. Introduction

A tile-based design methodology is not a new invention. It has been used in industry for many years, see e.g. [1][2]. However, there's no clear definition or description that is universally accepted. For example, in [1] a tile refers to an IEEE 1500 wrapped core. The authors of [1][3] show that such a core-based methodology has its advantages in particular with respect to logic test pattern generation. Similarly, we refer here to a tile as a design object, containing large quantities of logic, as well as embedded memories, both of which must be tested. However, the key difference that will become important here, is that the tiles we use have pass-through signals, allowing (jog-free) abutting of tiles in the layout, eliminating the need for any routing in the parent hierarchy level. Note that our tile-based notation here is driven from layout concepts; the physical location of a signal line (polygon) at the border of one tile matches up to the physical location of the very same signal line in the neighboring tile, and through this make a connection in the layout, without the need of a logical (and physical) port at the IOs of either tile. The latter usually implies some level of routing in the common parent instance, something that is not needed in our methodology.

Further differences are that each of our tiles do not have anchor logic near the ports, like an embedded TAP controller, IEEE 1500 WSP, or a pre-existing IEEE 1687 (IJTAG) host scan interface. Instead we expect the memory test insertion tool to add IEEE 1687 ports as needed, which we may also be used for other IJTAG compliant instruments in the tile. In addition, for the tile-based design methodology to work, all the used DFT tools must insert pass-through signal lines as needed.

The initial problem with the existing memory test tool is that it does not know about the tile-based design methodology. Instead, it follows a hierarchical, core-based design methodology. In this methodology, the DFT interface is inserted in each block, and expected to be connected only to a parent instance. An example is shown in Figure 1. This bottom-up methodology does not work in our design, as it leaves tiles unconnected, breaking the DFT signal connectivity. At the core of this paper, we describe how the used memory test tool can be taught to operate correctly for a tile-based design methodology, including the automation of insertion and connectivity of pass-through signals. The crux of the issue is automation, without which our DFT engineers would have to fix each and every tile connection manually, which is error-prone and time consuming.

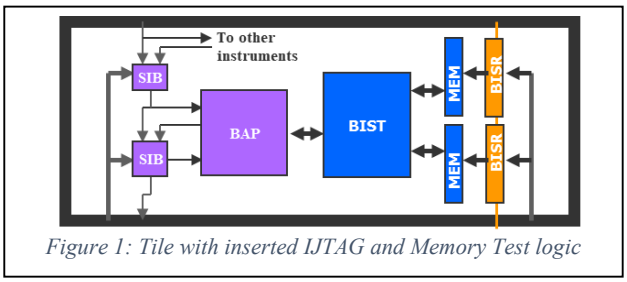

Figure 1: Tile with inserted IJTAG and Memory Test logic

In the next sections, we first review the typical hierarchical, core-based design methodology, and contrast it to the tile-based methodology we use for the production design, shortly outlined thereafter. However, let us first introduce the IJTAG components which form the backbone of the network connecting the memory test hardware components.

978-1-5386-5180-3/18/$31.00 ©2018 IEEE
DOI 10.1109/ITC-Asia.2018.00018