Design For Testability: Where does it Fit in the Design Flow?

Bulent Dervisoglu, Cadence Design Systems, 2670 Seely Avenue, San Jose, CA 95134

The traditional view of DFT paints it as an activity that is performed as part of Design Creation, which is executed during the front-end part of the design flow. This view has been supported by the following:

- The design cycle can be broken into the “Design Creation” (or logic design) phase followed by the “Design Implementation” (or physical design) phase. Logic synthesis is used to provide a mapping of the existing design description from a higher level of abstraction onto a lower level.
- DFT is inserted into a circuit by making structural changes to it. For example, scan and BIST are added by inserting special cells/circuits to enable the desired behavior during test. Since DFT is achieved via additions to the circuit description, DFT is considered as a part of Design Creation.
- The entire design (mission logic plus DFT circuits) can be processed by EDA software to transform the gate-level descriptions into a physical implementation (i.e. placement and routing), usually represented in the form of a GDS II tape.

The traditional approach has been possible since, until now, it has been practicable to achieve target design objectives by controlling certain optimization parameters during Logic Design/Synthesis that have first order of magnitude correlation with those objectives. For example, Area Optimization has been achieved through gate-level synthesis by minimizing gate count. Later, when operating frequency has become more important, Timing Driven Optimization has been achieved by considering switching delays of gates during the automatic synthesis process.

With the move towards smaller geometries, timing closure has come to depend a lot more on the distance that the signals have to travel so that the first order of magnitude correlation for timing closure has shifted from gate-delays to Placement, with Routing as the second order parameter (since better placement leads to better routing). Thus, performing synthesis without knowledge of how the synthesized elements are placed on the die has prevented making the right decisions that also maximize target system performance. This has opened the era for Physically Knowledgeable Synthesis (PKS) [1] as a paradigm for achieving area optimization simultaneously with timing optimization in high speed digital IC’s.

To achieve best results, the physically knowledgeable tool needs to recognize the DFT logic and separate that from the mission logic. This allows placement and synthesis decisions to be made primarily to optimize the mission logic and not be unduly affected by the connectivity of the DFT circuits. For example, existing scan chains need to be discovered and disconnected so that only the functional connections are taken into consideration during placement. Once this is achieved previous scan chains need to be reconnected using a new order for the flip-flops based on their placement. Thus, unlike the traditional design flow, DFT analysis and synthesis (e.g. re-order/re-connect scan chains) takes place during physical synthesis with PKS.

Today, certain DFT approaches, in particular scan and BIST, have become standard so that the only discussion is on how much, not what, to implement. EDA companies have developed tools that automate DFT insertion to such a level that the logic designer does not have to understand or know exactly what has been inserted in the design. Indeed, in some cases the designer may be unable to explain (or even care!) why certain design changes have been made. For example, a Logic BIST tool may make changes to the clocking structure of the flip-flops to prevent simultaneous launch and capture operations. The functional designer may be unable to fully understand why (or, even detect that) such circuit changes have been made. This makes it essential that the DFT tools understand why (or, even detect that) such circuit changes have been made. For example, a Logic BIST tool may make changes to the clocking structure of the flip-flops to prevent simultaneous launch and capture operations. The functional designer may be unable to fully understand why (or, even detect that) such circuit changes have been made. This makes it essential that the DFT tools understand why (or, even detect that) such circuit changes have been made. For example, a Logic BIST tool may make changes to the clocking structure of the flip-flops to prevent simultaneous launch and capture operations. The functional designer may be unable to fully understand why (or, even detect that) such circuit changes have been made. This makes it essential that the DFT tools understand why (or, even detect that) such circuit changes have been made.

Present design flows that use DFT insertion first during logic synthesis and a second time during physical synthesis may not produce best results for complex, high-performance designs. To assure increased optimization and shorten the overall IC design cycle it will be necessary to perform both logic synthesis and DFT insertion only once, using a physically knowledgeable synthesis tool.

References