TACOS: A Testability Allocation and Control System[*]

Laurent Ducerf-Bourbon[1] Peter Bukovjan[2] Meryem Marzouki[1†]

[1]LIP6 Laboratory, Couloir 55-65, 4 Place Jussieu, 75252 Paris Cedex 05, France
Laurent.Ducerf@lip6.fr Meryem.Marzouki@lip6.fr

[2]ON Semiconductor, B. Němcové 1720, 756 61 Rožnov p/R, Czech Republic
Peter.Bukovjan@onsemi.com

Abstract

This paper presents TACOS, a Testability Allocation and Control System. TACOS is a synthesis for testability environment which components can be integrated into a high-level synthesis flow. It transforms a VHDL RTL description of a data-path into its testable equivalent, supporting three testability methods: test point insertion, Scan and BIST. It also generates a test controller description, to be synthesized in the final circuit.

Keywords: Synthesis For Testability, High-Level DFT, Test Control

1 Introduction

TACOS is a tool-box, to be used by a designer in conjunction with a high-level synthesis system. Apart from its interactivity and versatility, this set of tools allows a high flexibility in its use, since it imposes neither a given testability methodology nor given test structure implementations. This paper details TACOS main components ans its use in conjunction with a HLS system.

2 IDAT module

The IDAT tool is the core of TACOS. It is able to modify a data-path architecture, described in the VHDL format, making it testable either by test point insertion or by allocation of Scan [3] or BIST [2] structures into the data-path, according to the user requirements.

3 Dynamic library of FUs

IDAT makes use of a library, which provides testable functional units (FU) when they are available, as well as additional test structures. Some of these components are static or fixed, but most of them are dynamically generated according to a parameterized description provided by IDAT.

Our choice is that IDAT should equally access to all these sorts of elements, by formulating a request determining the functionality and interfacing parameters of a given FU. IDAT should then be provided back with a description of the desired components in VHDL format.

The dynamic library is then organized as a database: an interface interprets a textual request and dispatches it to connected tools, running logic synthesis if necessary (to extract design characteristics like area, propagation delays, etc.) and converting the results in VHDL format. This interface also allows the use of estimators, which can be an alternative to running logic synthesis tools: for most of the dynamic components, mathematic formulas have been developed to estimate design characteristics, saving thus CPU time.

3.1 BSG module

When BIST methodology is focused, the constraint we have put is that a single TPG and a single TRC should be shared by all non-BIST FUs. Consequently, the TPG (resp. TRC) should be able to generate (resp. analyze) variable width data However, since IDAT operates at the RT-Level, the set of bit-width values of the FUs to consider is known and limited, allowing to avoid the definition of area-consuming fully reconfigurable test structures.

A BIST Structure Generator (BSG) has been de-

[*]This work has been partly supported by MEDEA-SMT Project, by a grant from the Grant Agency of Czech Republic, contract No. 102/98/1463 and by ON Semiconductor Czech Design Center in Rožnov p/R.

[†]Corresponding author

veloped and included in the dynamic library. As a starting point, the BSG allows the generation of reconfigurable Linear Feedback Shift Registers (LFSR) and Multiple Inputs Signature Registers (MISR). These elements are used in conjunction with multiplexers, parallel/serial converters, ROMs and comparators to define more complex TPG and TRC.

4 TesS module

In addition to the area overhead and the testability degree, IDAT takes into account test time induced by the generated testable architecture. This test time should be estimated in terms of number of clock cycles needed to reach an acceptable fault coverage using a given testability methodology and the test structures implementing it. This computation requires a dynamic analysis of the data-path architecture, and a specific tool is currently under development to meet this objective. The Test Scheduler (TesS) is hence in charge of two functions: test time estimation (running TesS concurrently with IDAT), and test plan generation (after IDAT completion), to be synthesized as the test controller of the final circuit.

5 TACOS integration

TACOS is now being integrated into a HLS system named UGH (User-Guided High Level Synthesis) [1], which is currently under development in our laboratory, in the framework of the European project ESPRIT COSY/OMI. UGH is an interactive HLS system for control-dominated co-processors, which target architecture is made up of a data-path controlled by a Finite State Machine (FSM).

The behavioral VHDL description is binded to a data-path template by CGS (Coarse Grain Scheduling), generating a coarse grain controller (the FSM) and specifying a fully functional data-path. The obtained data-path is mapped on a physically characterized target cell library, and the optimized controller is obtained by rescheduling the coarse grain FSM into FGS (Fine Grain Scheduling), taking into account the user specified target clock period and the data-path delays extracted from the physical cells.

Thanks to its modularity, TACOS can perfectly be integrated into UGH synthesis flow according to figure 1: IDAT produces a testable version of the fully specified data-path ; the test controller generation happens after the fonctionnal controller has been optimised (it will be also tested). The obtained test controller can then be optimised by FGS as well.

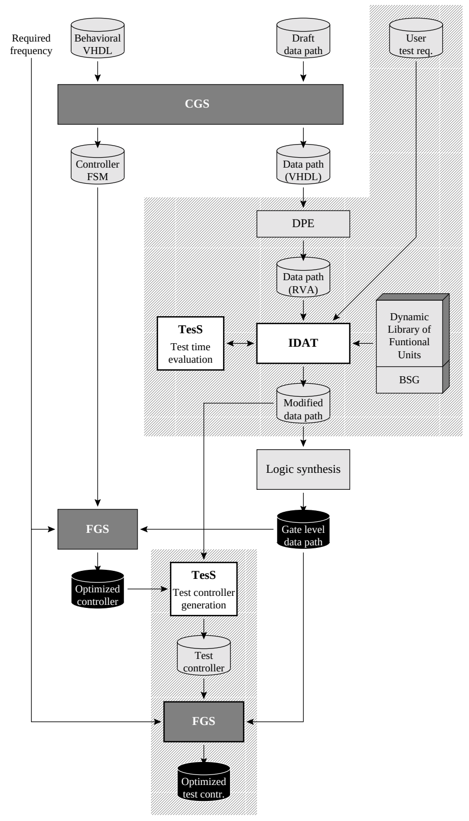

Figure 1: TACOS integration into UGH

6 Conclusion

TACOS allows to insert automatically test structures, relying on DFT reuse. This paper described briefly its components and an application into a HLS system that originally did not take testability into account. Further developments will be directed to test time evaluation and test plan generation.

References

[1] I. Augé, R. K. Bawa, P. Guerrier, A. Greiner, L. Jacomme, and F. Pétrot. User guided high level synthesis. In Ricardo Reis and Luc Claesen, editors, *VLSI: Integrated Systems on Silicon*, pages 464–475, Gramado, Brazil, August 1997. IFIP, Chapman & Hall.

[2] P. Bukovjan, L. Ducerf-Bourbon, and M. Marzouki. Cost/quality trade-off in synthesis for BIST. In *Proceedings of 1st IEEE Latin America Test Workshop (to appear)*, Rio de Janeiro (RJ), Brazil, March 2000.

[3] P. Bukovjan, L. Ducerf-Bourbon, and M. Marzouki. Cost/quality trade-off in synthesis for Scan. In *3rd IEEE Design and Diagnostics of Electronic Circuits and Systems Workshop*, Slomenice, Slovakia, April 2000.