Test Resource Partitioning: a Design & Test Issue

J.P. Teixeira[1], I.M. Teixeira[1], C.E. Pereira[2], O.P. Dias[1], J. Semião[1],
[1]IST / INESC, R. Alves Redol, 9, 1000-029 Lisboa, Portugal
[2]UFRGS, Porto Alegre, Brazil
jct@inesc.pt

ABSTRACT

Product development economics and specs drive the need for on chip embedded test functionality. However, optimal partitioning of test functionality between a tester and a SOC is a non-trivial task, which must be solved during the system analysis phase. Hence, at system level, a trade-off analysis must be performed, in order to evaluate the costs and benefits of different partitioning schemes. The purpose of this contribution is to present a methodology and tools, using the Object Oriented (OO) Paradigm and UML, and a set of architectural Quality Metrics (QMs), to analyze the impact of different TRP schemes on system's architecture. A 4-core SOC case study is presented to guide the discussion.

1. Introduction

DSM technologies, ever increasing complexity, performance and quality requirements, with shrinking time-to-market and product lifecycle put a heavy burden on new product development. Design productivity makes IP core design and test *reuse* mandatory, as well as standardization to allow multi-vendor core assembly in a SOC [1,2]. BST (BoundaryScan Test) interface and a standardized (P1500) wrapper interface (between each core and TAM (Test Access Mechanism) become necessary. Traditionally, test functionality was assigned to an external actor, the *tester* (or ATE, Automatic Test Equipment). As DUT complexity exponentially grows, the effectiveness of the DUT *external* test process decreases, and the ATE cost/complexity boosts. Major problems are the exploding test data volume and the bandwidth gap, i.e., the increasing separation between test data/time between the ATE and the SOC, and between the internal source/sink hardware and the CUT (Core Under Test). New applications require self test capabilities. Hence, due to test requirements, BITR (Built-In Test Resources) to implement a BIST approach are needed. However, the questions are: how much of the set (source, high-speed TAM, sink) infrastructure should be on chip, so affordable external ATE and lower speed TAM can be used? How can this be decided at specification phase, prior to implementation? Unquestionably, we face a design and test issue.

2. Methodology

Classic *structural* DFT techniques have been applied to enhance test effectiveness. TRP makes *system-level* DFT mandatory [3]. P1500 compatible core wrappers, TAMs and eventually core BIST are part of this strategy. A novel methodology (and tools) to generate, select and reconfigure hw/sw system architectures, based on a set of QMs (Quality Metrics) and decision criteria [3,4] is used to enable TRP trade-off analysis.

A cost-effective architectural solution implies autonomous, loosely coupled modules (objects, in OO semantics), as well as fully specified interfaces (in the temporal and value domain). The mission functionality (object *methods*) assigned to each object should lead to balanced object complexity (and execution time), and loose object associations. These characteristics are described by known sw QMs, such as *object cohesion*, *coupling* and *autonomy*. At specification level, the proposed methodology uses OO modeling techniques [5], together with the Unified Modeling Language. Extended DFDs, using a graph representation, describe *tasks* (methods) as graph nodes, and *attributes* as graph edges. Additionally, task and attribute *weights* are introduced. In this context, task *complexity* weights (w_{tc}) quantify relative task complexities. Task *time* weights (w_{tt}) quantify relative task execution time and the times the attributes up-dated by the task need to remain valid. *Attribute weights* (w_a) quantify their word length, as this has a direct impact on hw system complexity and can be used to address the bandwidth problem. In fact, low test pin count is modeled by larger w_a values. Architectural quality evaluation takes into account its network (modules and their interconnections) and its *dynamics* (the communication speed among modules in normal and test operation modes).

A target architecture is defined (`SysObj` tool) and system-level DFT is added by reconfiguration (`test-`

adder tool), adding on-chip test functionality to mission functionality (at present, BST, BIST or test point insertion). Thus, a *virtual boundary* between the ATE and the SOC can be defined (Fig.1), moving test functionality according to design criteria. This leads to different SOC architectural solutions, reflecting different object associations and communication speed among objects.

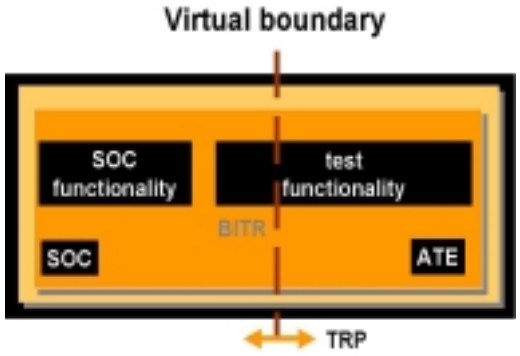

Fig. 1 - The concept of virtual boundary

Key QMs implemented in SysObj are shown in Fig. 2. Test-oriented QMs are the most adequate to deal with TRP, since they reflect the impact on system architecture and performance. System Task Overhead *(STOvh(w_{tc}))* is the % increase in system task weights, due to DFT. System Interconnection Overhead *(SIOvh(w_a))* is the % increase in the number of graph edges. Performance degradation is evaluated by the Critical Path Weight Overhead *(CPWOvh(w_{tt}))*, as the % increase in the sum of the weights of the tasks in the system critical path. Performance degradation is measured by CPWOvh and SIOvh, as this last metrics models the bandwidth problem at the SOC/ATE interface.

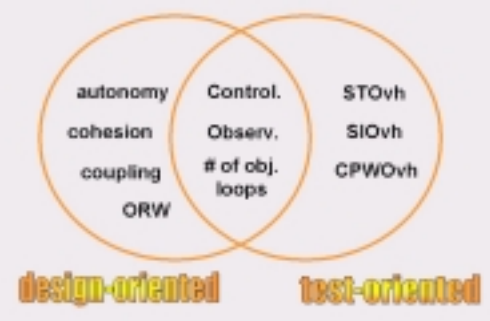

Fig. 2 - Key design and test-oriented QMs

3. Case-study

The case study is an electronic system for a real-time automation system, with 4 objects architecture, encapsulating 26 methods. For TRP purposes, six architectures are analyzed (see, e.g., archit. 3 in Fig. 3), all with BST wrappers, but with local (archit. 1,5,6) or global (archit. 2,3,4) TAP controllers, with (archit. 3,5) or without BIST, and with (archit. 6,4) or without enhanced w_a, to describe the bandwidth problem. Hence, the impact of the different options is decoupled.

Main results are summarized in table 1. BST wrapper insertion in the cores increases object autonomy from1 to 3, and induces around 3% STOvh (2% for a global TAP without BIST). BIST adds around 1% more in STOvh. A global TAP solution leads to a relevant (32%) CPWOvh (not shown in table 1). The bandwidth problem is measured as more than 200% in SIOvh (archit. 4 and 6), when source/sink test functionality is carried out by the ATE.

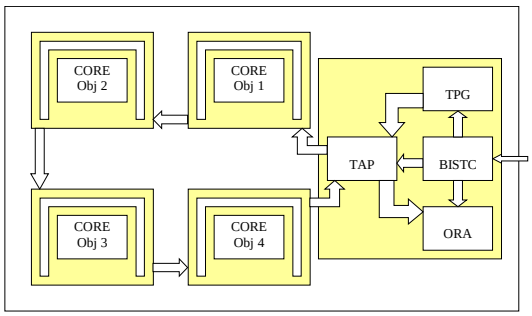

Fig. 3 - Archit. 3, serialized wrappers with common TAP, and BIST

Archit.	1,6	5	2,4	3
STOvh (%)	3.31	4.27	2.15	3.12
Archit.	**1**	**5**	**6**	
SIOvh (%)	128.6	133.3	235.7	
Archit.	**2**	**3**	**4**	
SIOvh (%)	121.4	126.2	228.6	

Table 1 - System Task/Interconnection Overhead

In conclusion, TRP is a design and test issue, that needs to be considered at system level. A methodology to support TRP trade-off analysis has been presented. Further work will include TAM modeling.

References

[1] R.K. Gupta, Y. Zorian, "Introducing Core-Based System Design", IEEE Design & Test of Computers, vol. 12, pp. 15-25, Sept./Oct. 1997.

[2] M. Keating, P. Bricaud, "Reuse Methodology Manual for Systems-on-a-Chip Designs", Kluwer, 1999.

[3] O.P. Dias, J. Semião, I.M. Teixeira, J.P. Teixeira, "TRP Quality Assessment: Trade-off Analysis", 1st. IEEE Workshop on TRP, Informal Digest, pp. 2.2.1-8, 2000.

[4] O.P. Dias, I.C. Teixeira, and J.P. Teixeira, "Metrics and Criteria for Quality Assessment of Testable Hw/Sw System Architectures", JETTA, vol. 14, nº. 1/2, pp. 149-158, Kluwer, Feb./April, 1999.

[5] J. Rumbaugh, et al., "Object-Oriented Modeling and Design", Prentice Hall, 1991.