Reuse-Based Test Planning for Core-Based Systems-on-Chip

Érika Cota[1*], **Marcelo Lubaszewski**[1,2]

[1]PPGC - Instituto de Informática
Universidade Federal do Rio Grande do Sul
Caixa Postal 15064 – 91501-970
Porto Alegre, RS

[2]PPGEE - Dept. de Engenharia Elétrica
Universidade Federal do Rio Grande do Sul
Av. Osvaldo Aranha, 103 – 90035-190
Porto Alegre, RS

erika@inf.ufrgs.br, luba@eletro.ufrgs.br

Abstract. *This work proposes two test planning heuristics that aim at reducing both the test costs of a core-based system and the computational effort to devise a cost-effective test plan. The proposed approaches target systems with distinct connection models, and the test planning algorithms are modeled accordingly. For systems connected through dedicated wires and buses, the test planning is modeled as a multi-variable optimization problem,and a search heuristic is implemented. For systems connected through a network-on-chip, the test planning is modeled and solved as a resource-constrained scheduling. Experimental results are presented for both techniques, and clearly show the variety of trade-offs that can be explored using the proposed reuse-based models, and the effectiveness of such methods on defining a cost-effective system test plan.*

1. Introduction

The density of current systems-on-chip (SoCs) and the paradigm of core-based design have posed important difficulties for the test of the resulting chip. As a result, testing has become one of the most expensive and time-consuming tasks of the circuit design, and may achieve 50% of the total cost of the chip [Zorian et al., 2000]. Therefore, the reduction of this cost is crucial for the electronic market.

One of the most important problems for the test of a SoC is the access to the embedded cores during test. Internal functional blocks require some type of electronic access during test, since they can not be directly accessed from the system interface. Such test access mechanism (TAM) is intimately related to the resulting testing costs of the system. For example, an exclusive access mechanism for each core results in a very reduced test time, but at a possibly unacceptable cost in terms of area overhead and pin count or, even, power consumption. Therefore, a great deal of effort has been spent in the last few years for the development of cost-effective test techniques for core-based SoCs, all of them aiming at the reduction of the test cost and complexity.

In this thesis, two test planning approaches focused on the reuse of the system resources during test and on the design space exploration are proposed. The test planning methods are defined according to the connection model of the SoC and aim at helping the system integrator and the test engineer to evaluate the impact of the test solution on the global system cost, for a number of possible configurations of a SoC.

In the first approach, a SoC with a core-to-core connection model is assumed and the test planning tool is based on two main aspects: 1) it considers a mixed set of access mechanisms; 2) both the

*Supported by Universidade Federal do Rio Grande do Sul, CNPq, and CAPES.

test schedule and the global test access mechanisms are defined together, through a multi-variable optimization algorithm. This expansion of concerns is combined with an efficient, yet fine-grained, search in the huge design space of the reuse-based environment so that good compromises among the various trade-offs being sought in the system synthesis can be found.

In the second approach, a system implemented over a communication platform called network-on-chip (NoC) is assumed. For those systems, the communication among the cores is implemented through a switching network that is integrated into the chip. This network already provides a real and efficient access to each block embedded into the circuit. Thus, instead of inserting new buses into the system with the sole purpose of test access, the reuse of the available communication platform during test is proposed and a test scheduling algorithm is developed. This is the first work that systematize the test of cores embedded into a NoC-based system.

The definition of a bus-based test access mechanism for embedded cores considering only the optimization of the resulting test time is an NP-complete problem [Chakrabarty, 2000]. A number of solutions for this problem have been proposed, many of them using ILP models, which can be very time-consuming. When other parameters, such as area, power, and reuse of available resources come into play, the problem becomes even more complex. Therefore, the solutions proposed in this work also consider the optimization of the computational effort required to devise a test plan, and the implemented heuristics have acceptable complexities, as it will be shown in Sections 3.3 and 4.1.

In this paper the proposed test planning approaches are introduced and some experimental results are presented. The context, motivation, and methods are detailed in the original thesis manuscript, which can be found at [Cota, 2004].

The sequel of this paper is divided as follows: Section 2 presents the main test requirements and test solutions for core-based SoCs. Sections 3 and 4 explain, respectively, the two reuse-based test planning approaches proposed in this thesis. Section 5 presents some experimental results. Section 6 concludes the paper.

2. Test Requirements and Related Works

One can divide the test requirements of the SoC in three levels: core requirements, interconnection requirements, and system requirements. The core requirements include the definition of the core test approach, the access to the core periphery and the core isolation during test. The test of the interconnections, in the second level, presents a single requirement: the possibility of precisely controlling and observing each connection. This test is, nevertheless, of extreme importance for the system characterization and test time. Finally, the system requirements include the definition of the test scheduling and of the test controller, and the combination of the access mechanism of each core in such a way that all cores are properly tested without deeply affecting the system performance, cost and design time.

Responding to the several test requirements listed above, one can group the test techniques presented so far in four categories: test access mechanism definition [Marinissen et al., 1998, Iyengar et al., 2001], test scheduling methods [Chakrabarty, 2000, Iyengar and Chakrabarty, 2002], test planing approaches [Iyengar et al., 2002], and standardization initiatives [Marinissen et al., 2002b, Marinissen et al., 2002a]. The techniques in the first group tackle the problem of defining an access mechanism to the cores periphery during test. Such solutions can either be based on the reuse of system functional connections, cores logic or on the insertion of a test bus. In the second group, the minimization of the test time is addressed, usually based on an access mechanism previously defined. In the third group one will find the techniques that take into account a number of system or test aspects. Finally, the fourth group comprises the initiatives for the standardization of the interface between the cores and the system during test, and the definition of a set of SoC test benchmarks.

The main drawback of the available solutions is that the test planning task is still conceived late in the design flow, when it is more difficult to change the system structure and constraints. Furthermore, each method assumes restricted TAM models on top of which the solution is searched (only test buses, only transparent paths, or only interconnections reuse). It is subsequently up to the designer to provide different amounts of test resources (as BIST controllers, TAM bitwidth, number of TAMs or test sets of the cores) and perform a number of searches under a variety of constraints. The test decisions, thus, are mostly based on the designer's experience, rather than on the characteristics of the system being designed.

Within this context, this thesis was defined to study a more comprehensive test planning method that can be inserted in the design cycle of the SoC as a support tool for the system integrator.

3. Test Planning and Design Space Exploration in Core-based Systems

This section presents a comprehensive test planning model for core-based systems, which privileges the reuse of system resources and aims at optimizing a number of cost factors. The main contributions of the proposed model are threefold:

1. it does not assume a single type of connection for the internal Test Access Mechanisms in the system. Partial test buses are considered, along with functional connections, transparency, and other bypass modes available through the wrapper or the core configuration;
2. the solution does not fully optimize every single core in the system. Instead, the diversity of test requirements among the cores is exploited, by privileging critical cores with more test resources;
3. both the schedule and the global TAM are defined together, and not as independent tasks as in other approaches. This aspect allows the exploration of the design space, so that good compromises among the various trade-offs being sought in the system synthesis can be found.

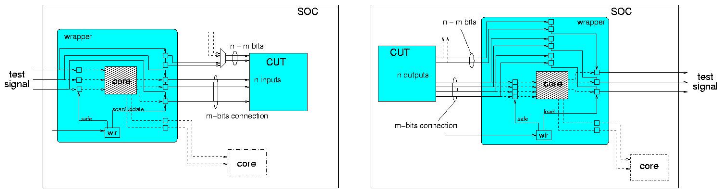
3.1. TAM Definition and System Cost Factors

We have defined a multiple yet not exhaustive model for connecting a core under test (CUT) to another core in the system (called here a *neighbor core*), or to the system periphery. This definition is based on the assumption that functional connections already available in the system can be reused during test to reduce test costs.

The complete access mechanism for any core in the system is defined as a series of connections between internal cores, so that two paths, one from the system interface to the CUT, and another one from the CUT to the system boundary are defined. Each pair of cores in the global TAM will be connected by one of the following methods:

1. reuse of available functional connections and functional pins of the next core in the path, as shown in Figure 1. In this case, only the wrapper of the neighbor core is reused;
2. use of the P1500 serial bypass [Marinissen et al., 2002b] in the next core in the path;
3. reuse only of the functional connections between the two cores;
4. use of available transparent modes in the neighbor core. In this case, the functionality of the neighbor core is reused to bypass the test signals.

In addition, any core can be connected directly to the system interface by means of an exclusive test bus. The TAMs were defined for scan, external (non-scan) and BIST testing schemes. Each one of the defined models represents an specific trade-off between the costs of pins, test time, area



(a) test inputs (b) test outputs

Figure 1: Access via functional connections

overhead, and power consumption at the system level. For example, the direct access to the system interface can be used for cores with few test pins requirements, or for cores that are close to the interface and present long testing times. On the other hand, serial bypass can be used by either the blocks with few test pin requirements or the ones with small number of test vectors. Thus, the best compromise among the test costs can be selected, based on the impact of the partial solution into the system and on the distinct requirements of each core in the chip.

3.2. Problem Statement

The problem of defining a global TAM and a test schedule for a SoC using the dedicated connection model can be stated as follows. Let the input be the set $C = \{i, 1 \leq i \leq N\}$ of cores, the set of test requirements for each core, the set $T = \{t_i | 1 \leq i \leq 5\}$ of possible test connection models between two cores (Section 3.1), the set of functions defining the costs for each test connection model, the graph of the functional connections among cores, the graph of distances between cores, the maximum power consumption (Po_{max}) allowed during test, and the maximum number of extra pins (Pi_{max}) that can be added to the system interface for testing purposes.

We want to find a solution comprised of:

1. A set of k_i-tuples $GT_{in} = \{(t_{a,b}, t_{b,c}, ..., t_{k,interface})_i, 1 \leq i \leq N\}$, for $t_{o,p} \in T$, and $o, p \in C$, representing the access path to the test inputs. The number k_i is the length of the TAM;
2. A set of k_i-tuples $GT_{out} = \{(t_{a,b}, t_{b,c}, ..., t_{k,interface})_i, 1 \leq i \leq N\}$, for $t_{o,p} \in T$, and $o, p \in C$, representing the access path to the test outputs. The number k_i is the length of the TAM;
3. a set $Times = \{(begin, end)_i, 1 \leq i \leq N\}$ representing the starting and ending testing cycles for each core i in the system test schedule.

Thus, the problem is to define a path from the system interface to the input and output of each embedded core and the system test schedule so that the total number of extra pins added to the circuit is less or equal to Pi_{max}, the power limit Po_{max} is respected at all times during the system test, and the total testing time and area overhead are optimized in this order.

3.3. Solution Modeling

Since the TAM is defined as a path, the modeling as a shortest path problem can be used. If we define the distance in the path as a function of the area, pins, power, and test time overhead for the core under test, traditional shortest path algorithms can be adapted to find the best path from the core periphery to the system interface.

Following this model one can define, for each core in the system, two trees: one for test inputs, and another one for test outputs. The root of each tree is the core under test (CUT), and the leaves represent the system interface. The algorithm for TAM definition consists, thus, in traversing the defined trees so that the shortest path from the root to the leaves is found.

However, the size of the trees considering all possibilities of connections is in the order of $N!$, and constructing and traversing such trees is impractical for most systems. Moreover, the search in each tree aims at optimizing the test costs for each core independently. Thus, the system global optimization can not be done if each tree is traversed without considering the status of the others. For example, if the best path for core 1 uses cores 2 and 3 before reaching the interface, the cost function of this path is related to core 1 only. This means that the test scheduling for the whole system must be defined after the definition of all test paths, so that cores sharing test resources are not tested at the same time. This kind of solution optimizes the resource allocation, but not the system testing time.

Hence, the shortest path search was adapted to tackle the aforementioned problems, and the solution was modeled as follows:

- the cost function of each arc between any two nodes in the tree is defined as a set of four values: time cost, area cost, pins cost and power cost;
- the tree is built on the fly, as the shortest path algorithm is implemented. Only promising nodes are expanded. The definition of a promising node is based on the available time slot extracted from the partial scheduling;
- the shortest-path algorithm is still used to traverse the tree as it is built, but testing time and power are not part of the path cost. Only pins and area are. Instead, time and power overhead are just checked in the partial scheduling so that previously established limits are respected;
- when a core a is used as part of the TAM for a CUT, a conflict is set between these two cores so that they are not tested in parallel. A conflict is also set between any other core using a and the CUT, since they are sharing a test resource.

In the thesis manuscript, the heuristic used to define the schedule and the resource allocation for the system is detailed, based on the modeling just proposed. The resulting test planning tool is called ReBaTe (**RE**use-**BA**sed **TE**st Planning). The complexity of the heuristic, also detailed in the manuscript, is $O(N^3)$, for N the number of cores in the SoC.

4. NoC-based Testing of Core-based Systems-on-chip

If a network-on-chip (NoC) is the communication platform of the system, the electronic access to each embedded core is available, since there is a real connection among all cores within the chip. The idea of reusing this resource during test is straightforward, since the access to the embedded cores is one of the main problems of the SoC test. If this reuse is possible, the pin and area overhead caused by the testing structures can be strongly reduced, for no dedicated TAM will be necessary. The main remaining problem is, thus, the test time.

In order to reuse the network-on-chip as TAM, the test vectors and test responses of each core must be expressed as a set of packets to be transmitted throughout the network. Then, the wrapper that connects the core to the network must be modified accordingly to send and receive the test data to/from the test interface of the core (scan controls, scan pins, functional pins).

One can consider that an external tester is connected to the functional system interface. Thus, the input and the output ports of the network can be re-used for the transmission of all packets (vectors and responses) to/from all cores. Let us also consider that only the cores and their corresponding wrappers are put into test mode, while the network routers and channels are kept in normal mode. This way, the same protocol used for functional communication is used during test. Notice that the test is still off-line, since the cores are in test mode.

The number of system interfaces used during test defines the initial number of paths that can be used in parallel to transmit test data. However, if there is more than one interface between the system and the external tester, the order on which the cores are tested is important, since different paths (and, consequently, different conflicts over the network resources) will be used for each core depending on the system input/output available at that time.

We propose a scheduling approach to systematize and automate the definition of test access paths through the network and a test sequence that optimizes the total test time. Considering the reuse of the functional system interfaces, the test costs in terms of pins can be drastically reduced. The area overhead caused by the modifications in the wrapper are also minimal.

4.1. Exploiting pipeline within the NoC

Let us consider the packets that must be transmitted through the network as the tasks to be scheduled, and the different access paths for each core as the resources that can be used by those tasks. Each task (packet) can use any available resource (path) capable of delivering the message to the correct destination. This means that different packets delivered to/from the same core can take different paths within the network.

There are two precedence rules for this problem. The first one is related to the test responses. A packet containing a test response can only be sent after the corresponding vector is completely received by the core and processed. Moreover, a new packet can be sent as early as an input path becomes available, as long as the response of the previous vector is not blocked. Thus, the scheduling of each packet must consider the possibility of conflicts and subsequent blocking of the output path, to avoid loss of data.

In the proposed approach, each channel in the network is assigned a time information, indicating when the channel is free to be used by a new packet. With this information, it is possible to schedule the next vector packet of a core as soon as one path is available, provided that the packet will not arrive at the core interface before the response payload can proceed in the network. This strategy combines the network and the core parallelism while ensuring that the internal scan chains are not overwritten.

The second precedence rule is more general, and deals with the priority of use of a given path. One can define that cores with larger number of packets and larger size of packets have priority to use shorter paths to reduce test time. The idea is to associate the shortest path to the most expensive core, to minimize its test time.

Finally, power consumption during test must be considered. The main advantage of the network reuse during test is the possibility of parallelization provided by this communication platform. However, as more cores are tested in parallel, the system power consumption during test may become an issue. In the proposed test technique, there are four sources of power consumption: the core, the wrapper, the router, and the communication channel. The power consumption model for each block was defined and incorporated into the test scheduling algorithm.

With these definitions, a variation of the list-scheduling algorithm was implemented. The resulting tool is called NoCBaTe (**NoC**-**BA**sed **TE**st Planning), and it is detailed in the thesis manuscript [Cota, 2004]. The complexity of the algorithm is $O[NlogN + N(C + T_0 * logT_0)]$, for N the total number of test packets, and T_0 the number of cycles of the resulting schedule.

5. Experimental Results and Discussion

Table 1 presents some comparative results between the two proposed test planning methods, for three SoC benchmarks [Marinissen et al., 2002a]. As the NoC-based technique only optimizes the system test time, this is the main parameter used for comparison. For the ReBaTe tool, the solutions with

Table 1: Comparative results between proposed approaches

Benchmark	Power limit	ReBaTe tool		NoCbaTe tool
		Test Pins	Test time	Grid 32-bit
h953	-	2	**223,139**	231,278
	50%	3	**218,391**	231,556
p22810	-	76	**281,626**	371,814
	50%	67	385,389	**371,814**
p34392	-	2	1,737,079	**764,570**
	50%	2	1,802,461	**834,060**

the smallest number of pins for the combinations of pins, time, and area optimization are considered in all cases, with or without power consumption control. For the NoCbaTe method, the solutions for the 32-bit grid topology are used. In Table 1, the smallest test times among the three possibilities are presented in bold.

It is interesting to notice that one method can be better than the other depending on the system configuration, and not only on the set of cores. This proves that the test planning tool must take into account the system characteristics, and not only the cores test requirements.

The results of Table 1 also demonstrate how the proposed test planning methods can be included in the beginning of the system design cycle. With the basic information about the cores the system integrator can define a possible floorplanning for the chip and use the ReBaTe tool to check the possible test costs for a core-to-core connection model. Similarly, the NoCBaTe tool can be executed assuming a communication platform is used. Thus, for example, if the designer is choosing between a bus-based or a NoC-based system, the test costs can be used as another parameter in the evaluation of both solutions, even before the system synthesis.

The comparison of the proposed methods to other SoC test solutions of the literature is not straightforward, since most techniques consider only a sub-set of the cost variables and TAMs used here. Nevertheless, one can find such a comparison in the thesis manuscript [Cota, 2004], under certain constraints. This comparison is not presented here because of the lack of space. It shows, however, that effective solutions can be devised even when more parameters are considered and non-dedicated TAMs are used.

6. Conclusions

This thesis presented two innovative approaches for the test planning of core-based systems. The first approach targets SoCs with a core-to-core connection model, whereas the second one targets NoC-based systems. For each connection architecture, the test planning problem was modeled, and heuristics were devised to efficiently search the space of test solutions and define test plans that are both generic and cost-effective, as required by a complex SoC.

This work presented a number of technical contributions, which were published in conferences and journals of impact. Two of them can be mentioned as example:

1. In the first test planning approach (ReBaTe), the whole system is considered and a solution that represents a good compromise among a number of costs is sought. This method was firstly published in [Cota et al., 2002b], when most papers on SoC testing were considering only the use of dedicated test buses. An improved version of the ReBaTe tool was published in the [Cota et al., 2002a], and was recently accepted to the *Journal of Electronic Testing: Theory and Applications (JETTA)*;

2. This is the first work that considers the test of heterogeneous NoC-based systems and reuses the available communication platform as test access mechanism. The method was firstly published in [Cota et al., 2003c]. The power-aware test scheduling was presented in [Cota et al., 2003a] and [Cota et al., 2003b]. Finally, a complete version of the method, considering other system configurations was accepted to the *ACM Transactions on Design and Automation of Electronic Systems*.

Despite the good performance of the proposed test planning techniques, there are some open points that can be studied in the sequel of this work, such as the consideration of other network topologies in the NoCbaTe method, and the expansion of the methods for mixed-signal SoCs.

References

Chakrabarty, K. (2000). Test Scheduling for Core-based Systems Using Mixed-integer Linear Programming. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 19(10):1163–74.

Cota, E. (2004). Thesis Homepage. Technical report. Available at: http://www.inf.ufrgs.br/ erika/Thesis/. Acessed in: March 2004.

Cota, E., Carro, L., Lubaszewski, M., and Orailoglu, A. (2002a). Generic and Detailed Search for TAM Definition in Core-based Systems. In *Latin American Test Workshop*, pages 160–164.

Cota, E., Carro, L., Orailoglu, A., and Lubaszewski, M. (2002b). Test Planning and Design Space Exploration in Core-based Environment. In *Design, Automation and Test in Europe*, pages 483–490, Paris, FR. Los Alamitos, IEEE Computer Society.

Cota, E., Carro, L., Wagner, F., and Lubaszewski, M. (2003a). Power-aware NoC Reuse on the Testing of Core-based Systems. In *European Test Workshop*, pages 123–128, Maastricht, NL.

Cota, E., Carro, L., Wagner, F., and Lubaszewski, M. (2003b). Power-aware NoC Reuse on the Testing of Core-based Systems. In *International Test Conference*, pages 612–621, Charlotte, USA. Los Alamitos, IEEE Computer Society.

Cota, E., Zeferino, C., Kreutz, M., Carro, L., Lubaszewski, M., and Susin, A. (2003c). The Impact of NoC Reuse on the Testing of Core-based Systems. In *IEEE VLSI Test Symposium*, pages 128–133, Napa Valley, USA. Los Alamitos, IEEE Computer Society.

Iyengar, V. and Chakrabarty, K. (2002). System-on-a-chip Test Scheduling With Precedence Relationships, Preemption, and Power Constraints. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 21(9):1088–1094.

Iyengar, V., Chakrabarty, K., and Marinissen, E. (2002). Wrapper/TAM Co-optimization, Constraint-driven Test Scheduling, and Tester Data Volume Reduction for SOCs. In *ACM/IEEE Design Automation Conference*, pages 685–690, New Orleans, USA. New York, ACM.

Iyengar, V., Chakrabarty, K., and Marinissen, E. J. (2001). Iterative Test Wrapper and Test Access Mechanism Co-optimization. In *International Test Conference*, pages 1023–1032, Baltimore, USA. Los Alamitos, IEEE Computer Society.

Marinissen, E., Arendsen, R., Bos, G., Dingemanse, H., Lousberg, M., and Wouters, C. (1998). A Structured and Scalable Mechanism for Test Access to Embedded Reusable Cores. In *International Test Conference*, pages 284–293. Los Alamitos, IEEE Computer Society.

Marinissen, E. J., Iyengar, V., and Chakrabarty, K. (2002a). A Set of Benchmarks for Modular Testing of SOCs. In *International Test Conference*, pages 521–528. Los Alamitos, IEEE Computer Society.

Marinissen, E. J., Kapur, R., Lousberg, M., McLaurin, T., Ricchetti, M., and Zorian, Y. (2002b). On IEEE P1500's Standard for Embedded Core Test. *Journal of Electronic Testing: Theory and Applications*, 18(4):365–383.

Zorian, Y., Dey, S., and Rodgers, M. (2000). Test of Future System-on-chips. In *IEEE/ACM International Conference on Computer-Aided Design*, pages 392–398.