Cost/Quality Trade-off in Synthesis for Scan*

Peter Bukovjan[1] Laurent Ducerf-Bourbon[2] Meryem Marzouki[2]
[1]ON Semiconductor, B. Němcové 1720, 756 61 Rožnov p/R, Czech Republics
Tel. (+420)651 667 147 - Fax. (+420)651 652 099
Peter.Bukovjan@onsemi.com
[2]LIP6/ASIM Laboratory, Couloir 55-65, 4 Place Jussieu, 75252 Paris Cedex 05, France
Tel. (+33)1 44 27 39 67 - Fax. (+33)1 44 27 72 80
Laurent.Ducerf@lip6.fr Meryem.Marzouki@lip6.fr

Abstract

This paper details our partial scan technique used in the Interactive Design for Test reuse in Allocation for Testability (IDAT) tool. IDAT tool objective is to fulfill the designer requirements regarding selected design and testability attributes of a circuit to be synthesized. A related tool is used to synthesize a test controller for the final testable circuit. The allocation process of scan resources in the data-path is not limited to registers, but also includes all sequential components existing in scan equivalent (design for test reuse concept). In order to optimize final test time multiple scan chain is also supported. Experiments show the results of our allocation for scan technique on two circuits.

1 Introduction

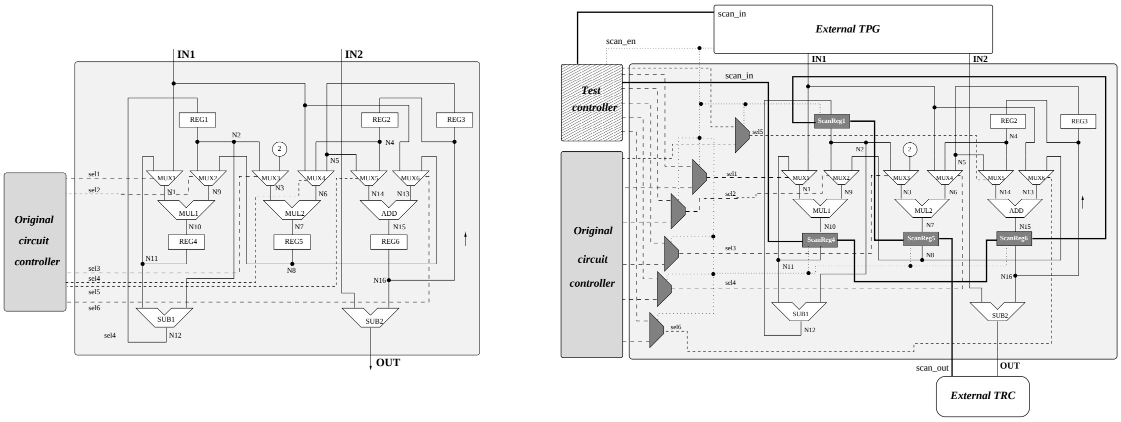
Partial scan is commonly used as a Design for Test (DFT) technique to improve the controllability and observability of circuit data-path internal nodes. Detailed background on different scan techniques can be found in [1].

In this paper we propose a new allocation for scan technique optimizing, from the testability point of view, the cost/quality trade-off during circuit synthesis from behavioral to Register-Transfer Level (RTL). The described technique is already implemented in the Interactive Design for test reuse in Allocation for Testability (IDAT) tool.

IDAT is used with a classical High-Level Synthesis (HLS) system to create an RTL circuit architecture optimized not only according to area and performance criteria, but also according to testability criteria. The whole allocation for testability process is viewed as the automatic generation of testable architecture starting from a draft RTL architecture enriched by selected design and testability attributes (this enriched architecture is called *Rich Virtual Architecture* (RVA)), and making profit of the DFT reuse concept. IDAT tool is able to modify existing data-path architecture in order to insert either test points, scan or BIST structures into the circuit and is complemented by a tool synthesizing the on-chip test controller, which implements the test plan (under development).

IDAT allocation for testability process consists of five steps: (1) extraction of the RVA architecture from the allocation step result of HLS system, (2) extraction of testable versions of allocated components from a library, (3) calculation of design and testability attributes for original circuit, where the data-path is already separated from the original circuit controller by a block of multiplexers, (4) cost/quality trade-off process itself, according to the testability methodology chosen by the designer and relying at each iteration step on the calculation of circuit attributes, (5) modified (testable) RTL data-path return to HLS system, in order to have precise scheduling be completed.

*This work has been partly supported by MEDEA AT403-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.



(a) (b)

Figure 1: Diffeq circuit: (a) original architecture, (b) scan architecture created by IDAT.

2 Scan architecture

An example of the draft RTL architecture taken as input by IDAT is depicted in figure 1(a). IDAT result, i.e. the resulting scan data-path architecture is shown in figure 1(b). Main features of our scan architecture are as follows: (1) All *select* inputs of the allocated multiplexers, all $RESET/SET$ inputs of sequential components and all registers *write enable control* inputs are connected through the multiplexers to the added test controller. (2) Scan transformation process is not limited to registers, but extended to all sequential FUs existing in library in its scan equivalent. (3) Multiple scan chain is supported.

3 Testability analysis for scan

In the case of scan methodology, the testability analysis process is used to guide the selection of the set of sequential components (registers, FUs), which should be transformed into scan versions. The general testability analysis approach has already been published in [4].

The purpose of the testability analysis algorithm - called propagation algorithm - is the calculation of the *controllability* (C) and the *observability* (O) of each node. The controllability (resp. observability) calculation for scan circuit starts both from primary inputs (resp. outputs) and scan components outputs (resp. inputs) and ends both at primary outputs (resp.inputs) and scan components inputs (resp. outputs). In current IDAT version, C and O reflect test problems caused by non-initializable feedback loops and transparency loose caused by traversing different FUs.

C and O values obtained for each node are then used to calculate the *controllability and observability exigencies* (CE and OE). These new metrics reflect the actual circuit test plan, where more than 90% of the test time is consumed by testing complex components, the FUs. The CE of a node corresponds to the necessity to control this node with respect to the test complexity of the components driven by this node. In the same way, OE of a node refers to the necessity to observe this node with respect to the test complexity of the component driving this node.

4 Scan cost/quality trade-off

The cost/quality trade-off process is the core of the allocation for testability process. The trade-off process has the purpose of making the best choice between various versions of a given component to be allocated, and to suggest inclusion of additional testability structures, in order to satisfy user requirements. In addition to user requirements, the trade-off algorithm takes also into account other criteria, including the mandatory requirements imposed by the chosen testability method, in our case scan. First of all, the trade-off process searches to satisfy the user requirements. If this is impossible with the existing library then a dialogue starts between IDAT and the user, who is asked to modify step by step his requirements, being guided by the system.

Some of already existing partial scan concepts are based on searching for the worst controllable and the worst observable nodes in the circuit in order to improve their C resp. O by replacing the selected registers with their scan equivalents. In our approach, we also take into account the possibility of existing scanable FUs in the data-path. This is done by our allocation for testability method based on the knowledge of CE and OE of each circuit node. CE and OE are strongly dependent on the complexity of components to which selected node is connected. In other words a complex FU, the inputs of which are not "well enough" controllable and/or which outputs are not "well enough" observable, either attracts allocation of scan resources in its close neighborhood or is directly transformed into its scan equivalent, when such an equivalent of this FU exists.

The allocation of a scan resource starts by the identification of a worst testable signal or a primary port and then compare their CE and OE:

- $CE \geq OE$: If identified node is the input of a FU, which can be transformed into a scan version, this FU is selected. If this node is connected to the input of more FUs, which can be transformed into scan versions, the FU with the biggest test complexity is selected. In other case a register or a FU, which can be transformed into a scan version and which drives the identified node is searched.
- $CE < OE$: If identified node is the output of a FU, which can be transformed into a scan version, this FU is selected. In other case a register or a FU, which can be transformed into a scan version and which is driven by the identified node is searched.

Found component is transformed into its scan equivalent and appropriately connected into the data-path. The decision on how many scan chains will be created depends on the trade-off between required maximal number of test cycles and minimal area overhead.

The trade-off is implemented with a branch-and-bound algorithm, using the previously described criteria. More details on the overall cost/quality trade-off during the allocation for testability process can be found in [3].

5 Results

5.1 Branch-and-bound example

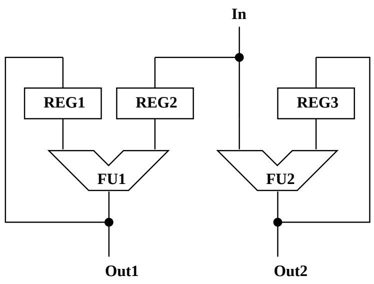
The branch-and-bound algorithm used is demonstrated on the following simple example. Let us consider the circuit in figure 2(a). Scan equivalents exist in the library for both FUs. FU1 area in scan version is bigger than FU2 area in scan version. Requirement for circuit testability is more than 90% and for area overhead is less than 11%. The branch-and-bound steps are depicted in table 1. This example demonstrates the situation when, despite existence of FU1 in scan version, it is more advantageous, from the final area overhead point of view, to test this FU by other means. This test is performed by transforming REG1 into a scan register, leading to a very small area overhead and by using primary input In. The problem is that sometimes, when tested FU contains many closed loops, such an "external" (with respect to the FU) test is not very efficient and it is necessary, in order to fulfill the testability requirement, to allocate a FU in its scan version.

step	**executed operation**	T	AO	**step**	**executed operation**	T	AO
1	replace FU1	$\underline{80\%}$	10%	5	return REG3 to its original	$\underline{80\%}$	10%
2	replace FU2	99%	$\underline{20\%}$	6	return FU1 to its original	$\underline{50\%}$	0%
3	return FU2 to its original	$\underline{80\%}$	10%	7	replace REG1	$\underline{60\%}$	2%
4	replace REG3	$\underline{85\%}$	$\underline{12\%}$	8	replace FU2	90%	10%

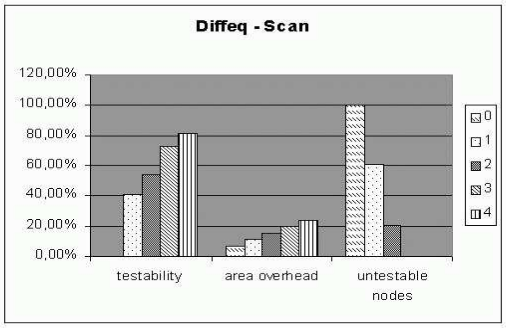
Table 1: Example of scan branch-and-bound session. Attributes T and AO correspond to testability and area overhead. Underlined values does not correspond to user requirements.

5.2 Diffeq benchmark

The benchmark Diffeq on which we will demonstrate our allocation for scan algorithm is originally described at the behavioral level. Its RTL data-path architecture is reprinted from [5] into



(a)



(b)

Figure 2: (a) Circuit example for scan branch-and-bound session, (b) selected resulting attributes (measurable in percents) for different scan circuit configurations of Diffeq.

figure 1(a). We have modified it by adding the necessary labels to components and signals, then we have converted the output comparator to subtractor SUB2 and finally we have connected this data-path to the circuit controller. The selected global attributes for each modification step performed by IDAT are depicted in the diagram of figure 2(b). User requirement for circuit testability is more than 77%, for area overhead is less than 25% and for untestable nodes is less than 3%.

Configuration 0 corresponds to the original circuit with allocated block of multiplexers. Because in the connected library of testable components only registers have their scan equivalents, only registers can be transformed into their testable versions. IDAT decided to make it in the following order: REG4, REG6, REG1 and REG5. Because user does not require any specific test time all above listed registers are connected into a single scan chain.

6 Conclusion

This paper has described the allocation for scan technique included in the allocation for testability tool IDAT based on cost/quality trade-off iteration algorithm, driven by a set of user requirements. Main advantages of the proposed allocation for scan technique, besides the foreseen short test application time, are: (1) the DFT reuse feature not limited to registers but extended to FUs existing in library in scan versions, (2) the avoidance of fault simulation necessity, (3) the consolidation, in a single tool, IDAT, of the synthesis of different testability methods (test point insertion, scan, BIST), (4) the use of the concept of "testability" instead of "fault coverage", in order to allow this consolidation of different testability methods in the same tool. The fault coverage itself is then measured by a combination of the testability percentage and the fault coverage allowed by a particular test methodology implementation. The benefit of this approach clearly appears when the chosen testability method is BIST [2].

References

[1] M. Abramovici, M.A. Breuer, and A.D. Friedman. *Digital System Testing and Testable Design, Revised Printing*. IEEE Press, NJ, USA, 1995. ISBN 0-7803-1062-4.

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

[3] P. Bukovjan, M. Marzouki, and W. Maroufi. Design for testability reuse in synthesis for testability. In *Proceedings of XII Symposium on Integrated Circuits and System Design*, pages 206–209, Natal, Brazil, September 1999.

[4] P. Bukovjan, M. Marzouki, and W. Maroufi. Testability analysis in high-level synthesis for testability. In *Proceedings from 10th European Workshop on Dependable Computing*, pages 71–75, Wien, Austria, May 1999.

[5] P.G. Paulin, J.P. Knight, and E.F. Girczyc. Hal: A multi-paradigm approach to automatic data path synthesis. In *Proceedings of the Design Automation Conference*, pages 263–270, June 1986.