Testability Analysis in High-Level Synthesis*

Peter Bukovjan Meryem Marzouki† Walid Maroufi

LIP6 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
Peter.Bukovjan@lip6.fr, Meryem.Marzouki@lip6.fr, Walid.Maroufi@lip6.fr

Abstract

This paper gives an overview of the testability analysis method which is part of our Interactive Design for Testability reuse in Allocation for Testability system IDAT. Our testability analysis process is based on the calculation of the controllability and observability for all nodes from a Rich Virtual Architecture description of a circuit. A global testability indicator is defined to describe the whole datapath testability. The paper details the testability definition, testability analysis algorithm, and then gives experimentation results.

Keywords: testability analysis, high-level synthesis for testability

1 Introduction

The testability measures presented in this paper are controllability and observability of a node together with global testability indicator defined to describe the whole datapath testability. Presented testability analysis is included in our system IDAT, developed to transform classical allocation step in a High-Level Synthesis (HLS) system to the allocation for testability step. In order to avoid an inaccurate and unpredictable mapping of the testability resources, and a loss of performances in the resulting testable architecture this system makes use of the virtual architecture of the circuit to be synthesized, which is neither the behavioral description, nor the RTL description.

Already existing testability analysis algorithm are not very convenient to be implemented to IDAT. The closest approache is Gu et al. testability analysis presented in [5], where controllability and observability of circuit nodes are calculated at RT level and then used to drive partial scan and test point insertion techniques. This analysis was later complemented by the definition of global testability indicator denoting the global testability of entire circuit [8]. Because Gu et al. do not suppose the creation of additional circuit test controller, their testability analysis needs to follow existing circuit controller instructions. Another close approach is Pires et al. testability analysis of the datapath from behavioral to RT level [3]. Their testability analysis results are used successively to drive register allocation, interconnection generation and finally to perform the test plan generation. Unfortunately this advantage is payed by loosing of testability measure fidelity, especialy the representation of sequential depth of the node, which is completely ignored in this approach. The testability analysis based on information theory was published in [7]. Information transmission coefficient (representing controllability measure) and mutual information transfer coefficient (core of the observability measure) are attributed to each circuit component and node using either sampling technique based on circuit functional simulation or composition technique suffering from ignorance of sequential node depth. In [2] the testability analysis technique for BIST at RT level is presented based on the measures of register randomness (related to register output controllability) and register transparency (corresponding to register input observability).

This article is a summary of one chapter from our detailed research report [1]. It is organized as follows: section 2 describes the design representation to which our testabilibility analysis method is applied. Section 3 details our testability definition, followed by the overview of our testability algorithm in section 4. Finally experimental results are presented in section 5.

2 Design representation

As was already stated, the testability analysis described in this paper is a part of our developed system IDAT. The goal of this system is to transform the resulting circuit datapath architecture from allocation and binding process of a High-Level Synthesis (HLS) system to a more testable one. This is made by using a library of already existing testable components (Design for Test reuse). The whole Allocation for Testability process is interactive and is driven by a

*This work has been partly supported by MEDEA-SMT Project and by a grant from the Grant Agency of Czech Republic, contract No. 102/98/1463.

†Corresponding author

test cost/quality trade-off algorithm.

The IDAT system makes use of the *Rich Virtual Architecture* (RVA) description format of the circuit. This RVA description is a draft RTL description of the circuit architecture, where the operators have already been allocated, and the variables have been assigned to registers. Contrarily to the RTL architecture, fine synchronization information are not yet taken into account. This RVA description (enriched with design attributes, like area, bit-width, maximal delay, etc.) is obtained after the allocation and binding steps of the HLS process.

3 Testability definition

Our testability analysis process is based on the calculation of the combinational and sequential controllability and observability factors. Although some concepts are common to other approaches presented in literature (especially to the approach of Gu [5], [4]), our testability analysis process differs from them by the fact that it operates on the virtual architecture and not on the true RTL one or on the CDFG, in addition to using different calculation formulas. In contrast to Gu's approach we suppose the creation of a circuit test controller, so in our testability analysis algorithm we do not need to follow the instructions from original circuit controller and we are limited only by existing datapath connections network (e.g. shared multiplexers *select* inputs).

The *combinational controllability* (CC) measures the difficulty to set a desired value to a component output node from the circuit primary inputs. To calculate this, it is necessary to know how difficult it is to set a desired value to a component output node from the component input nodes in the case this component is not embedded into a circuit. This difficulty is expressed by the component controllability transfer factor (CTF), which reflects the probability of setting a value at a component output by randomly exercising its inputs.

The *combinational observability* (CO) measures the difficulty to observe a desired value to a component input node from the circuit primary outputs. Like for CC, it is necessary to know how difficult it is to set a desired value to a component input node from the component output node by using of other component inputs in the case that component is not embedded into a circuit. This difficulty is expressed by the component observability transfer factor (OTF), which reflects the probability of observing a component input by randomly exercising its other inputs and observing its output.

The combinational factors range from 0 to 1.

The sequential factors - sequential controllability (SC) and sequential observability (SO) - range from 0 to infinity, and measure the sequential depth of the node, i.e. the estimated number of transition steps in our testability analysis algorithm to calculate the corresponding combinational value.

The controllability and observability attribute of a given node is calculated from corresponding combinational and sequential factor. After that, the testability of the node is calculated so as to undertake subsequent actions (allocation for testability provided by IDAT system) on hard-to-test nodes. To express the whole datapath testability the *global testability indicator* (T_{global}) was defined.

4 Analysis algorithm

A *propagation algorithm* has been developed in order to calculate the controllability and observability of the virtual architecture nodes. This algorithm is based on progressive calculation of the controllability and observability for each circuit node.

The controllability evaluation starts from the primary inputs and the scan components outputs (if any). Their combinational controllability (CC) is set to 1 and their sequential controllability (SC) is set to 0 (best possible values). All other nodes have their CC set to 0 and their SC set to infinity (worst possible values).

The controllability calculation then progresses towards the primary outputs and the scan components inputs, such that the worst values are modified only if the controllability of all the component inputs is greater than 0. However, this condition doesn't apply if the component is a multiplexer. For multiplexers, if the *select* input is not shared with other multiplexers, then the controllability of the multiplexer output is set to its best input controllability, otherwise, a *multiplexers output controllability trade-off algorithm* is executed, in order to find the higher average output controllability value of all the multiplexers sharing the *select* pin, and affect it to this input.

The observability calculation symmetrically starts from the primary outputs and the scan components inputs (if any). Their combinational observability (CO) is set to 1 and their sequential observability (SO) is set to 0 (best possible values). All other nodes have their CO set to 0 and their SO set to infinity (worst possible values).

The observability calculation then progresses backwards towards the primary inputs and the scan components outputs, such that the worst values are modified only when the inputs of the components on the path, other than the currently processed input, have the CC greater than 0. This condition doesn't apply when the component is a multiplexer.

5 Results

Our testability analysis process has been sucesfully applied to benchmark circuits like Diffeq [6]. The results will be reported in the final version of the paper because of lack of space in this summary.

In order to compare our testability analysis process with another approach presented in the literature and developed at Linköping University [4], we also have used the example depicted in figure 1, from [4]. Table 1 summarizes the comparison results. The correlation between results obtained with both approaches is quite high (except for SO), while indicating a slightly more preciseness of Gu's method, as shown in table 2. This slight difference is explained by the fact that Gu's method requires more knowledge on the synthesized circuit, namely information on the control part. One can note that our testability analysis process nearly reaches the same performances, by only working on the virtual architecture.

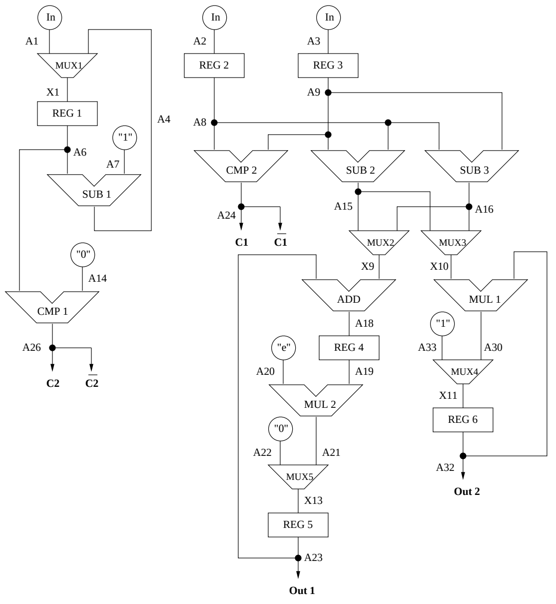

Figure 1: Example from Gu thesis [4]

Node	Gu's				IDAT			
	CC	SC	CO	SO	CC	SC	CO	SO
A1	1.00	0	0.55	5	1.00	0	0.36	3
A2	1.00	0	0.36	5	1.00	0	0.44	6
A3	1.00	0	0.36	5	1.00	0	0.44	6
A4	0.84	3	0.55	2	0.82	3	0.36	3
A6	1.00	3	0.55	2	1.00	2	0.36	1
A7	-	-	0.48	2	0.00	∞	0.31	4
A8	1.00	1	0.36	4	1.00	1	0.44	5
A9	1.00	1	0.36	4	1.00	1	0.44	5
A14	-	-	0.55	1	0.00	∞	0.36	1
A15	0.72	2	0.51	3	0.82	2	0.50	4
A16	0.72	2	0.51	3	0.82	2	0.50	4
A18	0.52	6	0.75	3	0.67	4	0.80	4
A19	0.52	7	0.75	2	0.67	5	0.80	3
A20	-	-	0.57	2	0.00	∞	0.53	3
A21	0.42	7	0.86	2	0.51	6	1.00	2
A22	-	-	0.86	5	0.00	∞	1.00	2
A23	0.36	9	1.00	0	0.51	7	1.00	0
A24	0.86	1	0.44	4	0.86	2	1.00	0
A26	0.86	4	0.86	1	0.86	3	1.00	0
A30	0.50	6	0.86	3	0.63	4	1.00	2
A32	0.43	9	1.00	0	0.63	6	1.00	0
A33	-	-	0.86	5	0.00	∞	1.00	2

Table 1: Comparison of Gu and IDAT testability analysis results for the *example*

CC	0.969819
SC	0.972457
CO	0.808689
SO	0.591362

Table 2: Correlation coefficients for the *example*

6 Conclusion

In this paper we have described the testability analysis module which is included in IDAT system, designed for improving of datapath testability. Our testability definition based on combinational and sequential factors of the node controllability and observability have been presented together with our testability analysis algorithm named propagation algorithm. Next steps are refinement of our algorithm for identification of the multiplexers with shared selection inputs and refinement of the formulas for calculation of controllability and observability together with corresponding changes in the propagation algorithm for different functional units.

References

[1] P. Bukovjan, M. Marzouki, and W. Maroufi. Allocation for testability by searching cost/quality trade-off in high-level synthesis for testability. Technical report, LIP6 Laboratory, Paris, February 1999.

[2] J. Carletta and C.A. Papachristou. Testability analysis and insertion for rtl circuit. In *Proceedings of IEEE International Conference on Computer Design*, 1995.

[3] M.L. Flottes, R. Pires, and B. Rouzeyre. Analyzing testability from behavioral to rt level. In *Proceedings of European Design and Test Conference*, pages 158–165, Paris, France, March 1997.

[4] X. Gu. *RT Level Testability Improvement by Testability Analysis and Transformations.* PhD thesis, Linkoping University, Sweden, 1996. LiU-Tek-Ph.D 414, ISBN 91-7871-654-3.

[5] X. Gu, K. Kuchcinski, and Z. Peng. Testability analysis and improvement from vhdl behavioral specification. In *Proceesings of the European Design Automation Conferenece with EURO-VHDL*, pages 644–649, 1994.

[6] 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.

[7] K. Thearling and J. Abraham. An easily computed functional level testability measure. In *International Test Conference*, 1989.

[8] T. Yang and Z. Peng. Incremental testability analysis for partial scan selection and design transformations. In *Proceedings of IEEE European Test Workshop*, pages 107–112, Barcelona, Spain, May 1998.