Cost/Quality Trade-off in Synthesis for BIST

P. Bukovjan (peter.bukovjan@onsemi.com)
ON Semiconductor, B. Němcové 1720, 756 61 Rožnov p/R, Czech Republic

L. Ducerf-Bourbon (laurent.ducerf@lip6.fr) and M. Marzouki (meryem.marzouki@lip6.fr)
LIP6 Laboratory, 4 Place Jussieu, 75252 Paris Cedex 05, France

Abstract. This paper details our allocation for Built-in Self Test (BIST) technique used by the core part of our Testability Allocation and Control System (TACOS) called IDAT. IDAT tool objective is to fulfill the designer requirements regarding selected design and testability attributes of a circuit data-path to be synthesized. A related tool is used to synthesize a test controller for the final testable circuit. The allocation process of BIST resources in the data-path is driven by two trade-off techniques performed in order to: (1) at the local level, select the optimal set of Functional Units (FUs) to be BISTed, using a new testability analysis method and (2) at the global level, for each selected FU of this set, choose either to allocate its BIST version (when available in a library) or to connect it to an internal Test Pattern Generator (TPG) and Test Results Checker (TRC). When necessary, a last step of the process is the allocation of scan chains used to test the remaining untested interconnections. Experiments show the results of our allocation for BIST technique on three benchmarks.

Keywords: BIST, Synthesis For Testability, DFT Reuse, Testability Analysis

1. Introduction

Built-In Self Test (BIST) techniques are based on the inclusion of test structures into a device so that it becomes able to test itself. This is achieved by the addition and corresponding connection of one or more internal Test Pattern Generators (TPGs), usually in the form of Linear Feedback Shift Registers (LFSRs) and one or more internal Test Result Checkers (TRCs) often in the form of Multiple-Input Signature Registers (MISRs). TPG and TRC can also be implemented in the form of Built-In Logic-Block Observers (BILBO) or CBILBO registers, which can work as both TPGs and TRCs. Historically, BIST techniques have concentrated on Functional Units (FUs) test, but some recent approaches also allow interconnection testing. Detailed background on BIST techniques can be found in [1].

Like all techniques attempting to improve the circuit testability, BIST techniques are used under the key constraint of reaching a required testability while keeping reasonable the area overhead and the test time. Approaches based on the use of BIST registers (usually BIL-

BOs) typically have the main objective of maximizing the sharing of these registers among tested blocks and minimizing their number (as for example in [12] or in [13]).

In [2] selected existing registers and FUs are used during the test phase as various TPGs and TRCs (LFSRs, arithmetic generators, etc.) to test remaining FUs. The extension of this approach to interconnection testing is achieved in [3]. This test is ensured by test point allocation. The disadvantage of this technique is the necessity of fault simulation to identify unobservable faults at each iteration step.

In this paper we propose a new allocation for BIST technique which optimizes, from the testability point of view, cost/quality trade-off relation during circuit synthesis. The described technique, which is already implemented in the data-path allocation for testability tool IDAT, takes into account the availability of a test controller, synthesized using a complementary tool called TesS (Test Scheduler). The methodology is based on two trade-off processes performed in order to:

– at the local level, select the optimal set of Functional Units (FUs) to be BISTed, using a new testability analysis method and

– at the global level, for each selected FU of this set, choose either to allocate its BIST version (when available in a library) or to connect it to a unique internal TPG and a unique TRC.

The proposed allocation for BIST process is iterative and interactive (driven by a set of users requirements). The TPG and TRC are shared by the FUs.

This paper is organized as follows. Section 2 gives us the overview of the whole Testability Allocation and Control System (TACOS) environment. Section 3 details the core of TACOS, namely the allocation for testability module IDAT. Description of IDAT module is, in section 4, followed by the description of BIST Structure Generator (BSG) module used, in the case of BIST methodology, to generate appropriate dynamically reconfigurable TPG and TRC when required. Performed modifications in order to transform original data-path into its BIST equivalent are detailed in section 5. Used testability analysis for BIST is presented in section 6, followed by the description of cost/quality trade-off for BIST algorithm in section 7. Obtained experimental results for three benchmark circuits are provided in section 8. Finally, the presented approach is summarized in section 9.

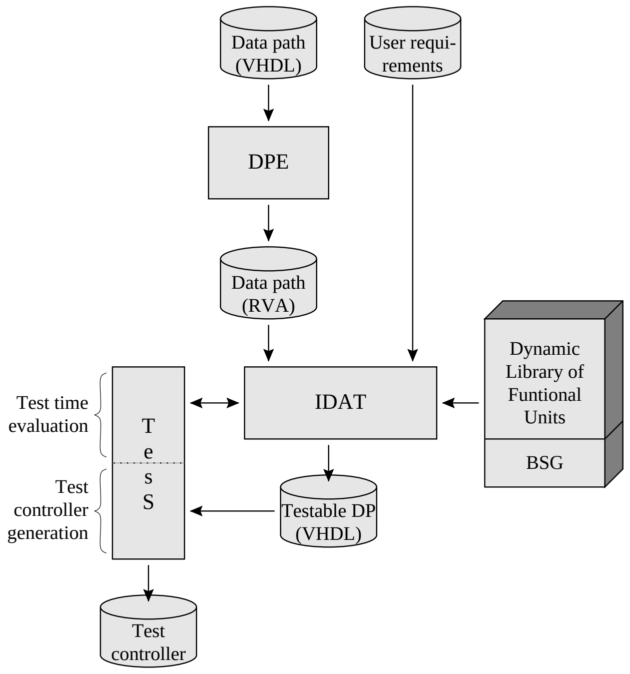

Figure 1. TACOS synoptic.

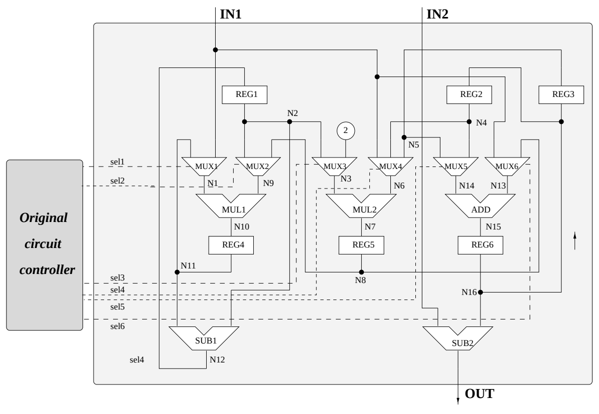

Figure 2. Original architecture of circuit Diffeq.

2. Overview of TACOS

TACOS, a Testability Allocation and Control System [11], is a kind of tool-box, to be used by a designer in conjunction with a high-level synthesis system (figure 1). Apart from its interactivity and versatility, this

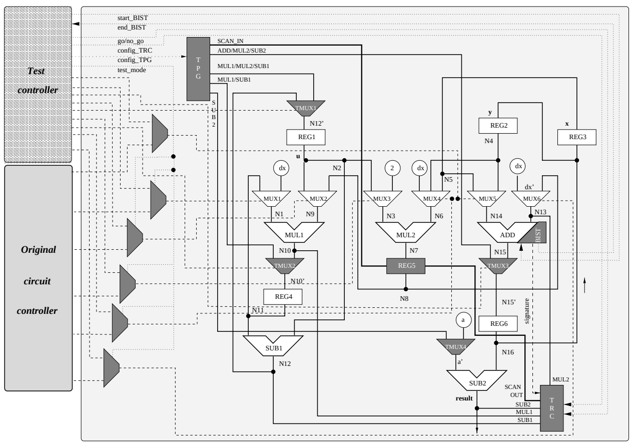

Figure 3. BIST architecture created by IDAT.

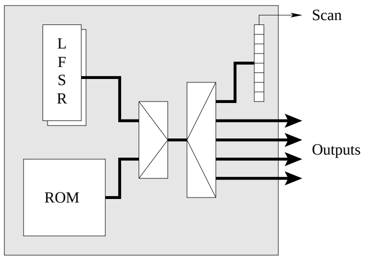

Figure 4. TPG architecture.

set of tools allows a high flexibility in its use, since it imposes neither a given testability methodology nor given test structure implementations.

TACOS is made up of four components: the Data Path Enrichment (DPE) module, the Interactive Design for Test reuse in Allocation for Testability (IDAT) tool, the Test Scheduler (TesS) and the Dynamic Library of Functional Units which includes BIST Structure Generators (BSG).

TACOS inputs are on the one hand a set of weighed user requirements regarding the design and testability attributes of the final circuits, and on the other hand a VHDL description of the initial (MUX

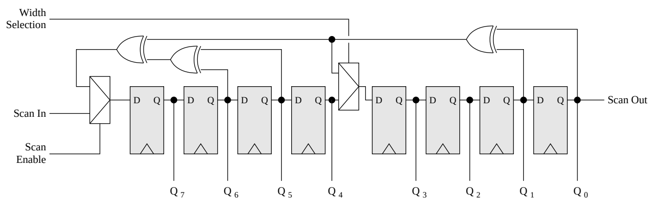

Figure 5. Example of a 8/4 bits reconfigurable LFSR.

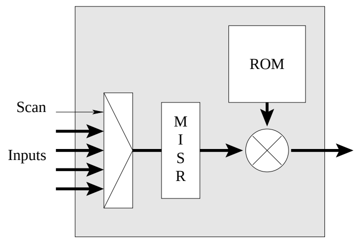

Figure 6. TRC architecture.

or BUS based) data-path of the circuit, at the RT-Level. The user can chose among three supported methodologies (test point insertion, Scan and BIST) to include the test structure into the circuit. The example of an input MUX based architecture at RT-Level is depicted in figure 2.

TACOS outputs the corresponding testable architecture (described in VHDL at the RT-Level), according to the user requirements in terms of testability attributes (e.g. area overhead, testability, etc.) and according to the chosen testability method. TACOS also outputs the test controller of the whole circuit, which is intended to test the functional controller as well.

Main TACOS steps are the following (see also figure 1):

1. The DPE module takes as input an initial data-path description in VHDL-RTL, and enriches it with some design and testability attributes of circuit components, which may be available in a library or obtained by running logic synthesis tools. Such an enriched architecture is called a "Rich Virtual Architecture" (RVA).

2. IDAT, step by step communicating with the user, converts a circuit data-path into its testable equivalent, making use of the dynamic library of testable components and the information about current data-path test time obtained from TesS. When BIST is focused, additional test structures may be necessary. The BSG module is able to generate to IDAT appropriate, dynamically reconfigurable, TPG and TRC.

3. Finally, TesS complements the testable data-path with a test controller to be added to the final synthesized circuit.

3. IDAT Module

The IDAT tool ([5] [6] [7] [4]) is the core of TACOS. It is able to modify a data-path architecture, described in the RVA format, making it testable either by test point insertion or by allocation of Scan or BIST structures into the data-path, according to the user requirements. The data-path allocation for testability concept for Scan and BIST is based on a *DFT reuse* of already testable components existing in a library and corresponding to original data-path components. In BIST mode, when such testable components do not exist, IDAT calls TPG and TRC structure generator (BSG), giving it the relevant parameters, in order to add TPG and TRC into the data-path.

Since a test controller is to be generated and added to the final circuit, the test of the data-path should be made independent from the original (functional) control part of the circuit, in order to get rid of the state dependencies of this control part during test sessions. This is achieved by identifying select signal of all multiplexers (resp. enable signal of all tristates), reset/set inputs of all sequential components and write-enable control input of all registers, and connecting these signals to the test controller through multiplexers (see figure 3). These multiplexers allow switching between normal and test modes. Such a method allows the use of a simple and quick testability analysis algorithm (improved version of the one presented in [8]), the reduction of external sequential ATPG algorithm complexity, and finally quick and straightforward test reaching high testability degree.

After this data-path isolation step through a block of multiplexers, IDAT extracts testable versions of allocated components from the library and starts the allocation for testability process itself, by making testability cost/quality trade-off, according to the chosen testability methodology and to the designer requirements. The cost/quality trade-off is an iterative process, guided at each iteration by testability analysis results and computation of design and testability attributes of modified circuit.

4. 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 selected non-BIST FUs. These FUs generally have different numbers of inputs and outputs. Consequently, the TPG (resp. TRC) should be able to generate (resp. analyze) variable width data, thus they need to be reconfigurable during one circuit test session. 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 developed and included in the dynamic library. As a starting point, the BSG allows the generation of LFSR and MISR. Figure 5 shows the internal structure of such a generated 8/4 bits reconfigurable LFSR. Generators of other kinds of test structures (e.g. based on cellular automata [9]) could also be developed or included in the dynamic library. These elements are used in conjunction with multiplexers, parallel/serial converters, ROMs and comparators to define more complex TPG and TRC (see figures 4 and 6).

5. BIST architecture

An example of the draft RTL architecture taken as input by TACOS is depicted in figure 2. TACOS result, i.e. the resulting BIST data-path architecture is shown in figure 3. Main features of our BIST architecture are the following:

– All *select* inputs of the allocated multiplexer (resp. *enable signal* of all tristates), all *RESET/SET* inputs of sequential components and all registers *write enable control* inputs are connected through the multiplexers to the added test controller.

– Already existing BIST versions of selected FUs are used to replace corresponding non testable FUs. This replacement in one hand improves test time and decreases complexity of main TPG and TRC if allocated, but in the other hand the resulting area overhead of BIST version is generally not negligible. When a BIST FU is allocated, all its test control signals are directly connected to the test controller.

– If a BIST version of a selected FU does not exist or if the area overhead resulting from the replacement of an original component by its BIST equivalent is too big, a unique TPG and a unique TRC are added to the data-path in order to test the selected components. All FU inputs (including control inputs) are directly or through one or more components connected to the TPG and to the TRC. The TPG and TRC are shared by all the non BIST FUs, thus they are parameterized and dynamically reconfigurable: the bit-width of TPG outputs and the bit-width of TRC inputs are modifiable. The BSG module (section 4) is used to generate such structures.

– Remaining untested interconnections can be tested by scan chains, starting from the TPG and ending at the TRC. In this case the *scan enable* inputs are connected together to the test controller.

6. Testability analysis for BIST

In case of BIST methodology, the testability analysis process is used to guide the selection of the set of FUs which should be BISTed. We use the improved version of the general testability analysis approach published in [8]. More details on the used testability analysis can be found in [4].

During a test session, the circuit data-path is controlled by the added test controller. The purpose of the testability analysis algorithm - called propagation algorithm - is the calculation of the *controllability and observability exigencies.* The controllability and observability exigency of a node is based on the *controllability* (C) and *observability* (O) of this node. The controllability (resp. observability) calculation for BIST circuit starts both from TPG outputs (resp. TRC inputs) and scan components outputs (resp. inputs) and ends both at TRC inputs (resp. TPG outputs) and scan components inputs (resp. outputs). In current IDAT version, the controllability and observability metrics reflect test problems caused by non-initializable feedback loops and transparency loss caused by traversing different FUs.

Obtained controllability and observability values for each node are then used to calculate the controllability and observability exigencies. 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 *controllability exigency* 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, the *observability exigency* of a node refers to the necessity to observe this node with respect to the test complexity of the component driving this node. These metrics, in our BIST technique, guide the selection of FUs which need to be tested.

As stated above, our testability analysis algorithm relies on analytical, calculated metrics. This allows to avoid the necessity of defining and applying actual test vectors after each data-path modification. The used method allows then to avoid a very complex and time consuming process, even if the test vectors are automatically generated and applied using an ATPG. The drawback is, obviously, some loss of accuracy.

7. BIST cost/quality trade-off

The cost/quality trade-off 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 the user requirements. In addition to these requirements, the trade-off algorithm also takes into account other criteria, including the mandatory requirements imposed by the chosen testability method, in our case BIST. Details on the case of Scan-based methodology can be found in [7].

The trade-off process searches first of all 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.

Testability problems are global in nature. A 100% testable component may show testability problems when it is embedded into a whole data-path. To avoid this situation, the trade-off algorithm is decomposed into two main modules: the *local trade-off* module, which acts at the component (FU) level, and the *global trade-off*, which operates at the data-path level.

During the local trade-off, the attributes of the original component and the attributes of each of its BIST versions are compared with corresponding user requirements from the user requirements vector. This results in a list of all versions ordered according to the level of satisfaction of user requirements. The global trade-off for BIST is made at two levels:

– Among all FUs in order to select the set of FUs for which the test will have a maximal effect on the overall circuit testability: the selection of a FU to test is guided by the testability analysis results and also, when more than one candidate with the same complexity exist, by the number of interconnections which can be tested together with this FU. A specific algorithm has been developed to select the longest possible test path and to reuse already allocated connections from TPG and to TRC, avoiding possible reconvergences (see figure 7).

  In figure 7, the selected FU to test is MUL1, and together with the test of MUL1 also the following interconnections are tested: TMUX1 $\rightarrow$ REG1, REG1 $\rightarrow$ MUX2, MUX2 $\rightarrow$ MUL1, TMUX2 $\rightarrow$ REG4, REG4 $\rightarrow$ MUX1 and MUX1 $\rightarrow$ MUL1. In IDAT current version the path reusing already existing connection has higher priority than the longest possible path.

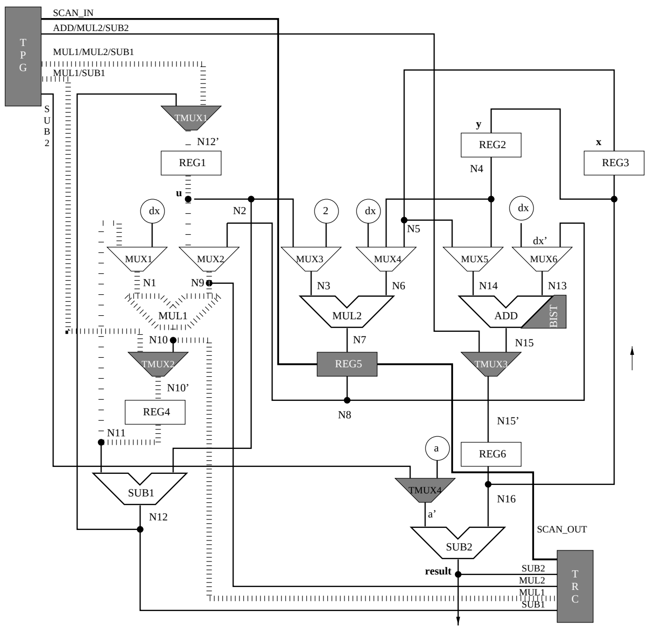

Figure 7. Data-path of the circuit from figure 3 with emphasized test paths to test MUL1.

– Between possibilities to either transform an original FU into its BIST version, when such a version exists, or to test the original FU using the reconfigurable TPG and TRC: the choice of a BIST version of an FU, compared to the connection of the original FU to a shared TPG and TRC usually decreases the test time of the circuit, reduces the TPG/TRC complexity and results in bigger area overhead. From the last reason, it is sometimes more advantageous to connect an FU to the TPG and TRC although its BIST version exists.

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 [7] and in [4].

8. Results

8.1. Branch-and-bound example

The branch-and-bound algorithm used is demonstrated on the following simple example. Let us have a circuit with four FUs which characteristics are summarized in table I. None of these FUs has a BIST version, so each of them, when selected to be tested, must be connected to the TPG and TRC. User requirements are as follows: testability 80%, area overhead 20%. The branch-and-bound algorithm steps are depicted in table II.

Table I. Example of four FUs to demonstrate BIST branch-and-bound algorithm.

FU name	% of global area	input #	output #
FU1	40%	16	8
FU2	20%	256	128
FU3	15%	16	8
FU4	15%	16	8

Table II. Steps of the BIST branch-and-bound algorithm.

BBA	executed	% of		Connected FU			
step	operation	T	AO	1	2	3	4
1	create TPG/TRC	30	5				
2	connect FU1	50	8	•			
3	connect FU2	70	19	•	•		
4	connect FU3	85	21	•	•	•	
5	disconnect FU3	75	19	•	•		
6	connect FU4	85	21	•	•		•
7	disconnect FU4	70	19	•	•		
8	disconnect FU2	50	8	•			
9	connect FU3	65	10	•		•	
10	connect FU4	80	12	•		•	•

The result of the branch-and-bound algorithm execution are: FU1, FU3 and FU4 connected to the TPG and TRC. This configuration satisfies user requirements of maximal circuit testability and minimal area overhead. Even though FU2 was first selected to be tested by the testability analysis process, this FU has finally not been connected to the TPG and TRC, because its high number of inputs and outputs results in too big area for the TPG and TRC (since both depend on the connected FU bit-widths), resulting in an unsatisfactory total area overhead.

8.2. Tseng benchmark

The first benchmark on which we will demonstrate our allocation for BIST algorithm is benchmark Tseng. Tseng is originally described at the behavioral level, while the input of IDAT is an RTL architecture. We have extracted data-path architecture from a simplified RTL one from [14] (see figure 8) and modified it by adding the necessary labels to components and nodes. The necessary design attributes were obtained by synthesis.

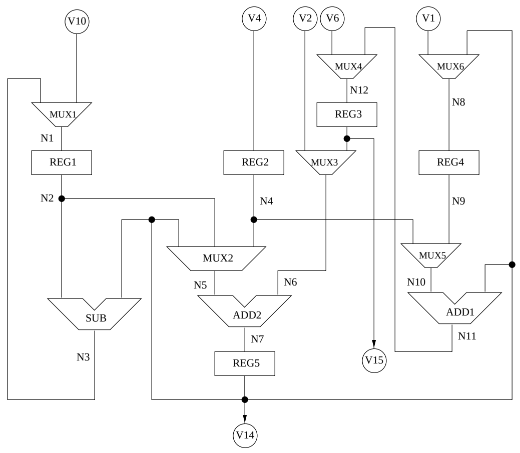

Figure 8. Data-path architecture of Tseng benchmark.

The evolution of the global attributes after each modification performed by IDAT is depicted in the diagram of figure 9. Each column with a different pattern, labeled from 1 to 4, refers to different circuit configuration number.

Configuration 1 corresponds to the original circuit with allocated block of multiplexers, separating the data-path from the original circuit controller. Other configurations correspond to the following step:

- (2) SUB selected: for SUB input N2, primary input V10 is connected to TPG, for SUB input, V14 signal N7 is connected to TPG, and SUB output N3 is directly connected to TRC,
- (3) ADD2 selected: for ADD2 input N5, signal N7 is already connected to TPG, for ADD2 input N6, primary input V6 is connected to TPG, and ADD2 output N7 is directly connected to TRC,

– (4) ADD1 selected: for ADD1 input N10 and ADD1 input V14, signal N7 is already connected to TPG, there is no direct reconvergence because signal N7 is connected through one register (REG5) to ADD1 input V14 and through two registers (REG5 and REG4) to ADD1 input N10 (with a different sequential depth), for ADD1 output N11, signal N6 is connected to TRC.

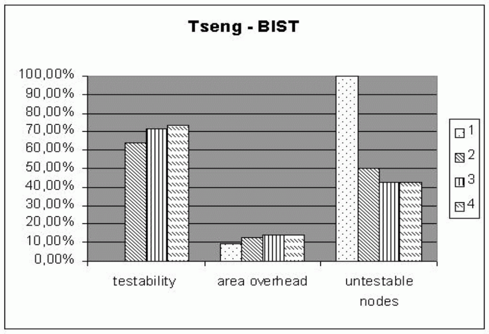

Figure 9. Selected resulting attributes (measurable in percents) for different circuit configurations of benchmark Tseng allocated for BIST.

8.3. Diffeq benchmark

The second benchmark on which we will demonstrate our allocation for BIST algorithm is Diffeq. Like Tseng benchmark, Diffeq benchmark is originally described at the behavioral level. Its RTL data-path architecture is reprinted from [14] into figure 2. 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 10.

Configuration 1 corresponds to the original circuit with allocated block of multiplexers. Subsequent modifications are as follows:

– (2) MUL1 selected: for MUL1 input N1, signal N10 is connected to TPG, for MUL1 input N9, signal N12 is connected to TPG, and MUL1 output N10 is directly connected to TRC,

– (3) MUL2 selected: for MUL2 input N3, signal N12 is already connected to TPG, for MUL2 input N6, signal N15 is connected to TPG, for MUL2 output, and signal N13 is connected to TRC,

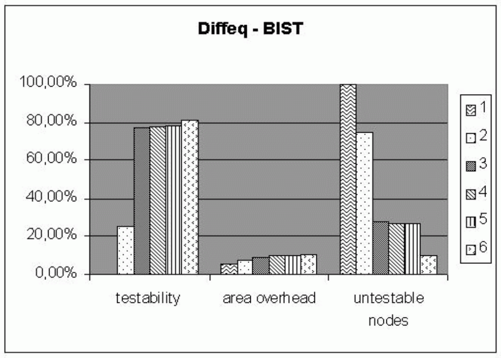

Figure 10. Selected resulting attributes (measurable in percents) for different circuit configurations of benchmark Diffeq allocated for BIST.

- (4) SUB2 selected: primary input IN2 is directly connected to TPG, for SUB2 input N16, signal N15 is already connected to TPG, and primary output OUT is directly connected to TRC,
- (5) SUB1 selected: for SUB1 input N11 signal, N10 is already connected to TPG, for SUB1 input N2, signal N12 is already connected to TPG, and SUB1 output N12 is directly connected to TRC,
- (6) ADD selected: for ADD input N14, signal N15 is already connected to TPG, for ADD input N13, primary input IN1 is connected to TPG, and ADD output N15 is directly connected to TRC.

8.4. 32-bits microprocessor MIPS R3000

The last circuit on which we will demonstrate our allocation for BIST algorithm is an in-house implementation of the 32-bits microprocessor MIPS R3000 (figure 11 [10]). We have modified it by adding the necessary labels to components and signals and we have connected this data-path to the circuit controller. Figure 13 shows the different connecting elements used to connect different signal types to TPG. When the signal is of type BUS, a tristate is used, otherwise a multiplexer is used. The selected global attributes for each modification step performed by IDAT are depicted in the diagram of figure 12.

Configuration 0 corresponds to the original circuit with allocated block of multiplexers. Subsequent modifications are as follows:

- ALU selected: ALU inputs X and Y are connected to TPG. ALU output DataOut is connected (through register AD) by Address to TRC,

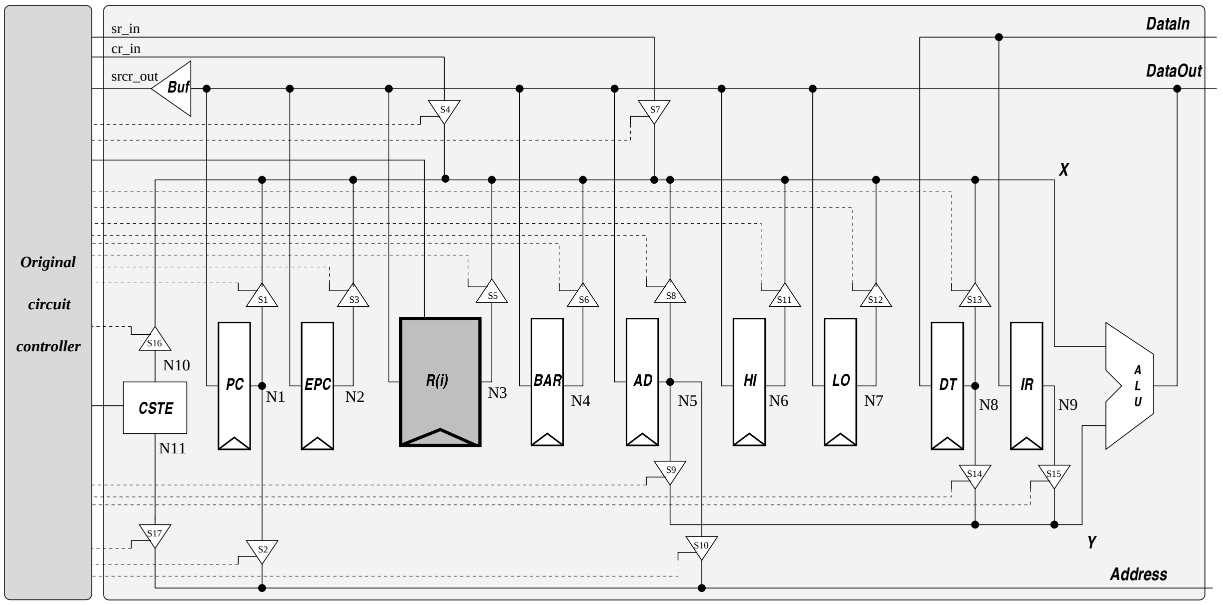

Figure 11. Original architecture of 32-bits microprocessor MIPS R3000

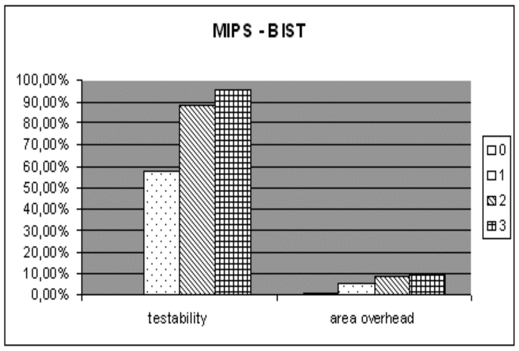

Figure 12. Selected resulting attributes (measurable in percents) for different circuit configurations of microprocessor MIPS allocated for BIST.

– register file R(i) selected: R(i) data input DataOut connected to TPG, R(i) address input connected to TPG. R(i) output X connected to TRC.

– registers DT and IR, which are not yet covered by the test, selected: registers input DataIn connected to TPG. Register output Y connected to TRC.

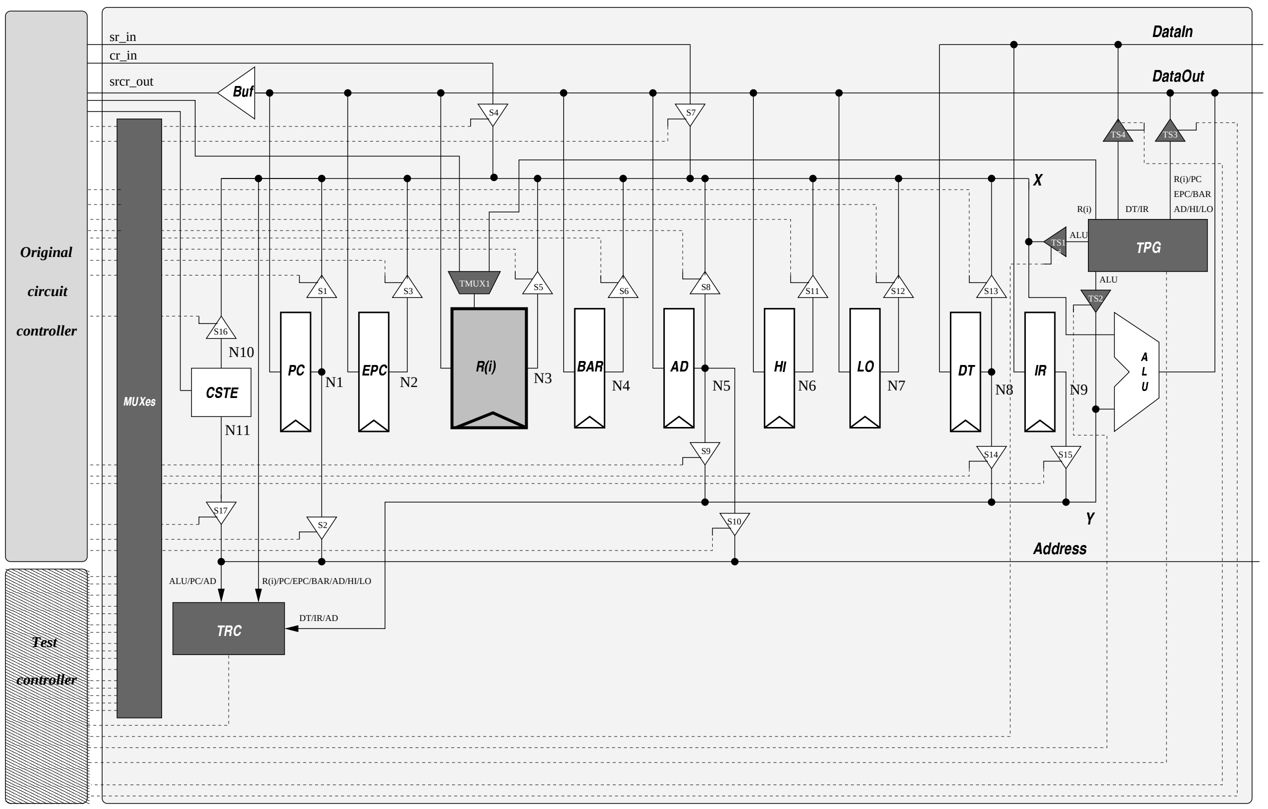

Figure 13. BIST architecture of MIPS

9. Conclusion

This paper has described the allocation for BIST 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 BIST technique, besides the foreseen short test application time, are:

1. the capability to test not only FUs, but also the interconnections,
2. the DFT reuse feature,
3. the minimization of added test structures, since a unique TPG and a unique TRC are shared by many FUs (to this purpose, the TPG and TRC are parameterized and dynamically reconfigurable),
4. the avoidance of fault simulation necessity, and
5. the consolidation, in a single tool, IDAT, of the synthesis of different testability methods (test point insertion, Scan, BIST).

10. Acknowledgments

Authors would like to thank Benoît Miramond for his results obtained in the framework of his DEA project "Processing of bus based architectures and memories in the allocation of test structures" and to Assan Saar for his implementation of BSG.

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.

References

1. Abramovici, M., M. Breuer, and A. Friedman: 1995, *Digital System Testing and Testable Design, Revised Printing.* NJ, USA: IEEE Press. ISBN 0-7803-1062-4.
2. Berthelot, D., M. Flottes, and B. Rouzeyre: 1998, 'OptiBIST: A Tool for BISTing Datapaths'. In: *Proceedings of IEEE European Test Workshop.* Barcelona, Spain, pp. 123–127.
3. Berthelot, D., M. Flottes, and B. Rouzeyre: 1999, 'Datapath BIST Scheme for Full Testing'. In: *Compendium of Papers from IEEE European Test Workshop.* Constance, Germany.
4. Bukovjan, P.: 2000, 'Allocation for Testability in High-Level Synthesis'. Ph.D. thesis, INP, Grenoble, France. http://www-asim.lip6.fr/publications/2000/index.gb.html.
5. Bukovjan, P., L. D. Bourbon, and M. Marzouki: 2000a, 'Cost/Quality Trade-Off in Synthesis for BIST'. In: *1st IEEE Latin America Test Workshop.* Rio de Janeiro, Brazil, pp. 110–115.
6. Bukovjan, P., L. D. Bourbon, and M. Marzouki: 2000b, 'Cost/Quality Trade-Off in Synthesis for Scan'. In: *Proceedings from Design and Diagnostics of Electronic Circuits and Systems Workshop.* Smolenica, Slovakia, pp. 30–33.
7. Bukovjan, P., M. Marzouki, and W. Maroufi: 1999a, 'Design for Testability Reuse in Synthesis for Testability'. In: *Proceedings of XII Symposium on Integrated Circuits and System Design.* Natal, Brazil, pp. 206–209.
8. Bukovjan, P., M. Marzouki, and W. Maroufi: 1999b, 'Testability Analysis in High-Level Synthesis for Testability'. In: *Proceedings from 10th European Workshop on Dependable Computing.* Wien, Austria, pp. 71–75.
9. Cardoso, P. S., M. Strum, J. de A. Amazonas, and W. Chau: 1999, 'Comparison between Quasi-Uniform Linear Cellular Automata and Linear Feedback Shift Registers as Test Pattern Generators for Built-In Self-Test Applications'. In: *Proceedings of XII Symposium on Integrated Circuits and System Design.* Natal, Brazil, pp. 198–200.
10. Dromard, F., Y. Body, M. Paget, A. Greiner, P. Bazargan-Sabet, and F. Pétrot: 1999, 'Interactive Learning of Processor Architecture'. In: *Proceedings of the 5th International Conference on Computer Aided Engineering Education.* Sofia, Bulgaria, pp. 123–129.
11. Ducerf-Bourbon, L., P. Bukovjan, and M. Marzouki: 2000, 'TACOS: A Testability Allocation and Control System'. In: *Compendium of Papers from IEEE European Test Workshop.* Cascais, Portugal.

12. Harmanani, H. and C. Papachristou: 1993, 'An Improved Method for RTL Synthesis with Testability TradeOffs'. In: *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*. pp. 30–35.
13. Parulkar, I., S. Gupta, and M. Breuer: 1995, 'Data Path Allocation for Synthesizing RTL Design with Low BIST Area Overhead'. In: *Proceedings of Design Automation Conference*. San Francisco, pp. 395–401.
14. Paulin, P., J. Knight, and E. Girczyc: 1986, 'HAL: A Multi-Paradigm Approach to Automatic Data Path Synthesis'. In: *Proceedings of the Design Automation Conference*. pp. 263–270.

Peter Bukovjan was born in 1971 in Bratislava, Slovakia. He received his Engineering degree from the Faculty of Electrical Engineering and Computer Science of Technical University in Brno, Czech republic and his Ph.D. degree from the National Polytechnical Institute of Grenoble, France, both in Microelectronics. He is currently with ON Semiconductor Czech Design Center where he is working in the group of Design System Technology. His research interests include design for test and CAD systems.

Laurent Ducerf-Bourbon was born in 1974 in Paray-le-Monial, France. He received the Masters Degree in Computer Science from the University of Besançon and the D.E.A Degree in Microelectronics from the University Pierre et Marie Curie in Paris. He is currently working towards his Ph.D. degree at LIP6 Laboratory under the direction of Meryem Marzouki. His research interests include fault coverage and test time evaluation for fonctionnal units based designs.

Meryem Marzouki was born in 1961 in Tunis, Tunisia. She received the D.E.A Degree and the Ph.D. Degree, both in Computer Science, from the National Polytechnical Institute of Grenoble in 1987 and 1991 respectively. Before that, she received the Engineering Degree in Computer Science too, from the University of Tunis, in 1986. From 1987 to 1992, she was mainly involved in prototype validation of ICs in the framework of electron-beam testing. Since 1992, she has been a researcher with the CNRS (French National Research Center), in charge of the *Diagnosis of Complex System* Group at TIMA Laboratory in Grenoble, France, with artificial intelligence approaches for test and diagnosis as main research interests, until 1997, when she joined the LIP6 Laboratory in Paris, where she is in charge of the Test group. Since then, her main research interests are related to high-level synthesis for testability and system on a chip testing.