Les cahiers Leibniz

Toward an Optimal Scan Insertion at the Register Transfer Level

Lilia Zaourar, Yann Kieffer, Nadia Brauner, Chouki Aktouf

Laboratoire G-SCOP
46 av. Félix Viallet, 38000 GRENOBLE, France
ISSN : 1298-020X

n^o 186

September 2010

Site internet : http://www.g-scop.inpg.fr/CahiersLeibniz/

Toward an Optimal Scan Insertion at the Register Transfer Level

Lilia Zaourar, Yann Kieffer, Nadia brauner
Lilia.Zaourar, @g-scop.inpg.fr,
Yann.Kieffer@g-scop.inpg.fr
Nadia.Brauner@g-scop.inpg.fr
Laboratoire G-SCOP/ INPG 46 Avenue Félix Viallet
38031 cedex 1 Grenoble France

Chouki Aktouf, Vincent Juliard
Chouki.Aktouf@defactotech.com
Vincent.Juliard@defactotech.com
DeFacTo Technologies, 167 rue de Mayoussard, 38430,
Moirans France

Abstract:
This work shows improvements towards effective consideration of scan pre-synthesis. A new algorithm is proposed to optimally decide about scan stitching at RTL. Algorithm efficiency is proven through an industrial RTL-to-GDS design flow and typical design benchmarks.

Keywords:
DFT, RTL, Scan, TSP, Graph Models.

1. Introduction

As the increasing complexity of designs is stretching the limits of existing tools, a move to a higher level of abstraction helps to address the challenges posed by traditional DFT solutions. The most cost-effective approach – in terms of engineering resources and time-to-market – is to start performing accurate design analysis as early as possible in the design cycle. As design complexity increases, today and into the future, each step in the design process must be accomplished at a higher level of design representation. The requirement for a higher level of abstraction includes all design steps and techniques, including DFT, as long as quality is not compromised. Low-level design steps should focus only on final back-end design steps and optimizations such as those for power timing and floor planning, which remain affordable even as designs get larger.

Today, for complex ASIC designs, scan is a mainstream DFT methodology which is considered during the design cycle. Full scan methodology is most used. Design functional Flip-Flops are usually replaced by scan Flip-Flops and subsequently scan chains are built. Connecting all sequential elements adds a significant amount of wirings which may cause congestion problems during placement and routing, and also needs additional logic, meaning additional area and more delays in the design.
Usually, not much can be done for the additional logic, which is inherent to “full scan”. To reduce wire length induced by scan chains and hence limit the impact of scan insertion on the final layout, scan chain optimization is required.

To benefit from RTL or high-level scan, scan-chain optimization should be realized based on the information which is extracted from the RTL design. The main challenge here is that no restitching is possible – since the scan chain gets mixed during synthesis with the design functional logic.

Previous research works on the scan optimization topic analyzed logic around FFs including data path in between [3, 16]. Other work focused on ordering between FFs either at gate [4, 5, 6, 7, 8, 9, 10] or higher levels [2]. For some work on delay fault testing at RTL [17], it was shown that moving scan insertion back to design time does not prevent any fine tuning of scan, as needed for delay-fault testing.
In this paper, the proposed work does not consider any kind of design analysis, except for the extraction of memory elements (FFs) and the expected proximity between FFs. Hence as detailed later, the overall scan insertion methodology remains simple. Indeed it takes fully advantage of synthesis tools where both functional and scan logic are jointly synthesized. The main concept of the proposed methodology was originally proposed in [1]. In this work, the real focus is on a mathematical graph model which leads to using a standard algorithm for the TSP problem.

This paper is organized as follows. In section 2, main motivations around RTL scan insertion, basic methodology and overall characteristics are reviewed. In section 3 and 4 both the problem and the mathematical model are defined. Afterwards, section 5 describes the proposed method for optimizing scan ordering. Finally, experimental results are presented before concluding the paper.

2. Motivation

Traditional gate-level scan leads to costly iterations around logic synthesis. Indeed, during both scan insertion and the ATPG process, several changes are usually required back to RTL (see Figure 1). Furthermore, traditional scan is also not compliant with new design methodologies. IP based design becomes a common practice. Most of exchanged IP cores are reusable or soft, i.e. synthesizable. DFT-friendly IP do not offer any guarantee that testability is solved.

Consequently, testability enhancement process becomes more and more complicated since accomplished after IP design and also because it is made during or after synthesis

Any extension or customization of gate-level designs with scan becomes very difficult with the complexity of designs. Indeed, beyond classical use of scan for manufacturing testing, Scan logic needs to be extended to cover new needs such as yield learning, design (e.g. power-on reset) and security modes. Also, the verification process of a complex gate-level design scan becomes a bottleneck. Finally, a reusable design methodology is meaningful only if all the logic that is related to DFT, including scan, is implemented at the same level where main design decisions are made -- that is, at RTL.

Adopting scan at RTL strengthens verification policies and consolidates reusable design methodologies by closing the gap between RTL design and design for test. Beyond the natural benefits of synthesizing both application and scan logic (the ability to decrease the area overhead of test logic, reach better optimization in terms of design performance), the existence of RTL scan solutions naturally will decrease the verification effort.
Consequently, two different flows have been considered to illustrate the proposed methodology and validate the experimental results.

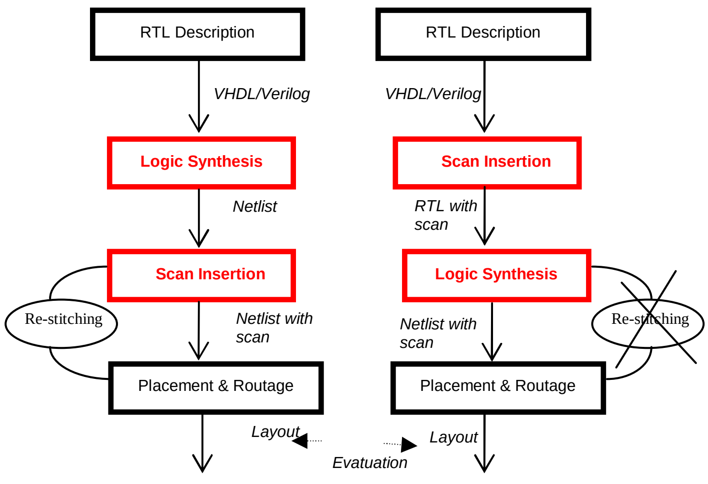


Fig. 1 scan at the gate level Fig. 2 scan at the RT level

The first flow is a traditional flow, where scan logic is inserted after the logic synthesis process [Fig. 1]. The insertion process is improved while iterating on the placement & routing step by re-stitching.

The second flow is based on RTL scan insertion [Fig. 2]. In RTL-scan, the RTL code (in Verilog or VHDL) is generated by the scan-insertion tool where scan test logic is added to the original RTL code. This code is then synthesized, offering the possibility to optimize these added RTL constructs together with those which are related to the functional design. The resulting netlist serves as an input during placement & routing. In this flow, it is not expected to redesign, or rerun scan stitching, once both synthesis and testability issues are solved.

This research work focuses on the ordering problem, while scan insertion at RTL is doing using the methodology of [2]. Example 1 illustrates the method. For more details see [2].

```
module dff(ck, rst_n, din, qout
// HIDFT PORT
 , se , si , so);
// HIDFT ADD_INTERFACE
input se, si ;
output so ;
// HIDFT ADD_NET
input   ck, rst_n, din;
output  qout;
reg     qout;
always @(posedge ck or negedge rst_n)
  if(~rst_n) qout <= 1'b0;
// HIDFT ADD_SCAN_ASSIGN
else
if ( se )  qout <=  si ;
 else  qout <= din;
// HIDFT ADD_CONTINUOUS_ASSIGN
assign so = {qout} ;
endmodule
```

Figure 3: Example for RTL_scan Insertion with Hidft_tool

With an RTL scan methodology, it is noteworthy that ATPG tools continue to be used as traditionally with not methodology changes. Main changes are related to scan re-stitching which is become not possible since functionally, the design behaves as if scanned FFs where present. But the FFs in the design are purely functional FFs. Hence, re-stitching is not possible mainly because synthesis do not ensure mapping RTL scan logic into scan FFs.

3. Problem position

Scan insertion has a significant impact on chip performance like area overhead, function and test power. The main purpose is to find an optimized ordering of FFs and minimize overall impact.

A heuristic-based analysis is considered. Such heuristic illustrates the results on the above parameters when scan logic is inserted at RTL. It is important to notice that cell area increase due to logic insertion depends mainly on the relationship in the design between consecutive FFs in a scan chain. Power consumption during test is mainly reduced by ATPG tools whose process comes later in the flow. Mainly, ATPG tools take advantage of “don't care” bits to minimize the switching activity during scan test. Timing is usually not negatively impacted since timing closure is realized once the scan insertion process is terminated. Dynamic power during normal use of the design is related to capacitances, and thus to wire length. Finally, congestion during routing decreases when additional wire length is reduced. Hence, a special focus is given to an optimal scan stitching solution with minimum global wire length.

4. Mathematical model

We now present a mathematical model for the construction of optimal scan chains from the RTL description. This model is based on graph theory. First, we derive a graph that represents the design. We deduce from this graph a second one that expresses some form of proximity between the memory elements.

4-1 Lightweight synthesis

Since the RTL description of the design may be behavorial, abstracting away from gates, FFs and nets, we need first to translate the RTL into these elements. This is what synthesis does, but we don't want to spend the time and effort to do a real synthesis step at that point. Once RTL-scan is inserted, the synthesis step has to be done anyway, and we don't want to duplicate that effort.

Our solution is to do a *leightweight synthesis* as the first step of the RTL scan insertion. This synthesis is done in terms of generic gates; it does not try to optimize logic, timing or gate sizes; it does not need the user to set any parameters; in short, it is done transparently, and the user never gets to see the netlist that results from it: from the standpoint of the user, the input to the RTL-scan insertion tool is the RTL.
From this netlist we now define the Design-graph (in short *D-graph*) G_D.

4-2 Building the D- graph

The undirected D-graph G_D is constructed as follows (see Figure 5):
- Each memory element, gate and net from the design corresponds to a vertex in graph G_D.
- There is an edge between two vertices if and only if the corresponding elements in the RTL description or in the net list are connected.
Graph G_D is undirected because it only aims at giving indications on how memory elements might be located at the placement step.

4.3 P- graph for memory elements proximity

The second step is to extract information from the design that is necessary for scan chain stitching optimization. This information will be given in the form of an edge-valuated Proximity Graph (in short P-graph), to which we will now refer to as G_P. The P-graph G_P will be built from the graph G_D as follows:
- The vertices are the memory elements of the design,
- There is an edge between two vertices if there is an electronic connection in the design between the memory elements they represent.
- The edges are valuated by some notion of expected proximity that we call functional proximity. The weight of edge (i,j) is denoted by w_{ij}.

Definition 1:
Functional proximity measures the `logical distance' between two memory elements. It is evaluated by a shortest non-oriented path between the corresponding vertices in the graph G_D.

If two memory elements are far apart, it is not meaningful that one follows the other in the chain ordering. Therefore, we fixed beforehand a threshold t, which is a limit on the length of a path between two memory elements. If all paths between two flip-flops are longer than t, the corresponding vertices are not connected in P-graph G_P.
Hence G_P is computed from G_D by taking the memory elements as vertices, and inserting an edge between two vertices if and only if the memory elements are electronically connected and there is a path between them not greater than t in G_D. Note that the paths considered for defining the edges of G_P are not allowed to go through a memory element. The weight of the edges, w_{ij} will be used later during the optimization step for scan chain construction.

4.3 Algorithm for the construction of G_P

The actual computations for determining the weights w_{ij} of the edges of G_P are done using Breadth-First Search (BFS) [13] on G_D. The BFS algorithm performs a breadth-first traversal that visits vertices that are closer to the source before visiting vertices that are further away. In G_D, we do a BFS algorithm on all vertices corresponding to memory elements. In this context “distance” is defined as the number of gates on the path from the source vertex to any other memory element.
In this work, we adapt the standard BFS algorithm in order to limit the depth to t. Therefore we consider only the nodes at a depth smaller that the limit value t.
The computation time for this algorithm is bounded by the product of the number of edges in G_D by the number of flip-flops of the design.
While in the whole method, this phase can be seen as some kind of pre-treatment, it is noticeable that some optimization already occurs at this step, through the search for shortest paths between

memory elements. These computations help devising a precise set of data for the forthcoming optimization step.
Example1 illustrates the construction of G_D and G_P.

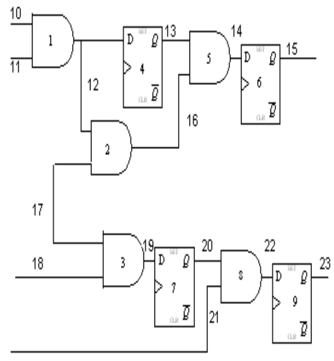
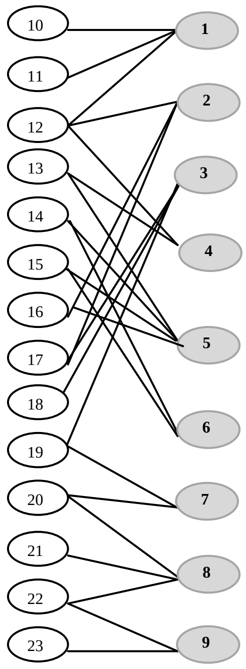


Figure 4 : Design example 1



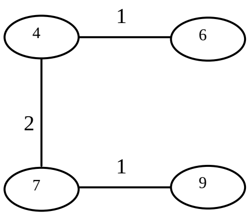


Figure 6 : Graph G_P

Figure 5 : Graph G_D

4.4 Mapping the scan chain problem to a Hamiltonian path

The Graph G_P represents the electronic connections between memory elements together with their functional proximity. The scan chain stitching problem can now be modeled as a classical graph theory problem.
Scan Chain Problem:

- Instance *:*

$G_P = (V_P, E_P)$ undirected, with:
V_P = {memory elements}
$e_i \in E_P$: electronic connexion between two vertices v_i et v_j
w_{ij}: the length of the edge e_{ij} representing the functional proximity

- A feasible solution :

A chain of size n that passes exactly once through all the vertices of G_P

- Objective:

A chain with minimum length of edges.

Finding a good stitching for scan chain memory elements at RTL therefore translates mathematically to finding a minimum-weight *Hamiltonian path* in the graph G_P built in the previous section.

Definition 2:
A **Hamiltonian path** is a path that goes through all vertices of the graph exactly once.

5. Scan stitching algorithm

Given a RTL design, the main idea consists in extracting information from the RTL description on the proximity of memory elements.

Scan stitching problem is now equivalent to finding a path that goes through all the memory elements that are the vertices V_P of the graph G_P. This reduces to a classical problem in graph theory which is the Traveling Salesman Problem (TSP) [12,13].

Definition 3:
In the general form of a **Traveling Salesman Problem (TSP)**, we are given a complete graph $G=(V, E)$ and a cost C_{uv} for travelling between each pair (u, v) of vertices of V. A tour is a path that passes exactly once through each vertex in V and returns to the starting vertex. The objective of the Traveling Salesman Problem is to find a tour of minimal cost.

In our case, to construct a scan chain ordering, we do not need to return to the starting point: a Hamiltonian path is enough. Moreover, graph G_P is not complete.
Hence, our problem can be reduced to the TSP by adding all missing edges in graph G_P with a huge weight and by adding a vertex that is adjacent to all vertices of the graph. Then, removing this vertex from an optimal tour provides an optimal Hamiltonian path. We choose to put huge weight to added edges to force the algorithm to avoid them: theses edges would correspond to added wire in the circuit. Indeed, our objective is to minimize the global wire length including added wire.
Although this problem is NP-Complete [14], there exist polynomial-time approximation algorithms to solve this problem.

Definition 4 :
An ***α*-approximation algorithm** for a minimization problem is an algorithm which output is at most α times the optimal value thus guaranteeing a worst case performance of α.

There are well-known approximation algorithms for the TSP. Considering memory space and running time, we have chosen the classical 2-approximation algorithm to find an ordering of the memory elements that minimizes the additional wire length of scan chains. The 2-approximation algorithm starts by finding a minimum spanning trees T which then goes through a shortening procedure. This procedure considers a path that goes through all vertices, perhaps several times, and returns a shorter Hamiltonian path.
Since, a complete graph on n vertices has $n(n-1)/2$ edges, memory space considerations limit the size

of the studied designs to thousands of memory elements.

In the worst case, this algorithm could give a solution that is twice the optimal. However, in practical cases, the solution is often rather good. Indeed, we have compared this algorithm to an optimal one on several circuits and the solution was always less than 10% more than the optimal.

Figure 7, show how we insert this algorithm to Hidft -scan tool.

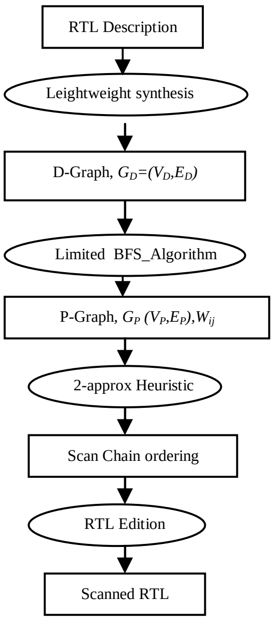


Figure 7 : RTL Scan Insertion

6. Implementation and Experimental results

The proposed method has been validated using several designs both in VHDL and Verilog. The performance is based on the wire length. The computation time is also given for the evaluation of the method. The designs for the experimentations are from academic and industrial origin: ITC99 and open cores. First experimentations were conducted on several test cases ranging from small to large numbers of flip-flops with single clock designs from ITC99 benchmarks in VHDL. Other experimentations on the following Verilog open cores were also done: simple_spi which is a SPI core, Biquad which is a DSP core and Ac97 a controller core. For more details about these open cores see [16]. All the algorithms are implemented in C++. For the sake of clarity we present here only part of the test cases we considered: those having the most significant results.

The performance is measured on the total wire length of a chip obtained after placement and routing from the proposed scan chain ordering at the RTL. We compare these results with the total wire length obtained before synthesis at the gate level. Scan at the gate level is inserted by Magma's integrated flow; scan at the RTL is inserted with HiDFT-Scan from DeFacTo technologies. Fault Coverage was verified with the ATPG tool Tetramax to verify that the fault coverage remains correct.

TEST_ NAME	#ff	#gates	G_DVer	G_DEdg	G_PVer	G_PEdg	CPU_ G_DG_P	Time 2-aprox
				ITC99(VHDL)				
b09	28	131	401	719	28	378	<1ms	<1ms
b10	17	172	525	947	17	136	10ms	<1ms
b11	31	336	1094	1845	31	465	20ms	10ms
b12	121	1000	3879	6997	121	7260	200ms	20ms
b13	53	309	633	1042	53	1378	20ms	<1ms
b14	245	3461	27269	46778	245	29890	3s	110ms
b15	449	6931	33179	56798	449	100576	6s	320ms
b17	1415	21191	100358	171759	1415	697990	30s	3s
b18	3320	49293	271320	462326	3320	2330318	3min	10s
b19	6642	98726	531161	906201	6642	7061713	5min	20s
				Open Cores (Verilog)				
Simple-spi	132	7565	1427	2320	132	8515	100ms	20ms
Biquad	204	4097	3757	5835	204	20706	350ms	80ms
Ac97	2289	18348	18707	30490	2289	2196067	30s	8s

Table 1 : Graph descriptions

Table 1 indicates the size of the objects on which we work from Section 4.3. The first column gives the name of each design. Then, the number of flip-flop and gates in the circuits are given followed by the sizes of the graphs (number of vertices and number of edges). The last two columns indicate the CPU time for the construction of G_P and for the 2-approximation algorithm. The depth of the BFS is limited to t = 4. Note that the number of vertices in graph G1 is exactly the number of flip-flop of the design.

TEST_NAME	# ff	WL_Gate	WL_RTL	slack %
b09	28	2,06	1,63	-20,59
b10	17	1,94	2,06	5,97
b11	31	4,77	5,03	5,38
b12	121	12,60	12,92	2,59
b13	53	3,39	3,32	-2,25
b14	245	63,64	64,24	0,96
b15	449	126,32	122,04	-3,39
b17	1415	395,62	370,41	-6,37
b18	3320	1187,15	922,55	-22,29
b19	6642	2329,63	1858,19	-20,24
Simple-spi	132	11,46	10,43	-8,95
Biquad	204	34,10	31,33	-8,14
Ac97	2289	247,70	238,52	-3,71

Table 2 : Total Wire length after scan insertion

Table 2 describes the total wire length including original interconnections for each test case for scan gate level and for RTL scan. The last column gives the ratio between the proportional difference for the wire length between the RTL scan and the gate level scan.

Table 1 indicates that the running time for finding the optimal scan stitching is sufficiently small to be included in a complete flow. Indeed even for circuits with more than 10 000 gates, the complete scanning process takes less than 10 minutes. Table 2 shows that the proposed method reduces the total wire length as compared to the gate level for large circuits by around 20%. For small circuits there might be a loss limited to 6%.
Table 3 gives the fault coverage obtained by Tetramax. It shows that the fault coverage remains good.
Finally, scan at the RTL does not degrade the total wire length of the layout. These numbers would be indicative of the results that can be expected for larger designs.

Test_Name	b09	b10	b11	b12	b13	b14	b15
Fau_Cov	99.86%	99.85%	99.93%	99.97%	99.92%	99.99%	99.97%
Test_Name	**b17**	**b18**	**b19**	**simple_spi**	**Biquad**	**ac97**	
Fau_Cov	99,4.7%	99,81	99,81	98,35%	99,96%	99,80%	

Table 3 : Fault Coverage

Conclusion and perspectives

This work contributed in taking into account wire length as a concrete parameter to optimize during the RTL scan insertion process. A graph theoretic approach with the related mathematical model was presented. The obtained results demonstrated that the increase in wire length due to RTL scan insertion is close to traditional gate-level scan. Wire length was measured after a placement and routing design step. Typically, first results, for more than 10 000 gates give a wire length equivalent between scan gate and RTL scan.
Finally, this method is reusable and technology independent.

To extend this work and give a broader support for the argument that scan can be inserted at RTL; it would be worthwhile to investigate the effective behaviour of timing and power under this method. It is also possible to improve the proposed mathematical model to take into account other parameters like clock domains. It is also planned to validate this method for much complex designs.

References

[1] C. Aktouf, H. Fleury, and C. Robach. “Inserting scan at the behavioral level” *IEEE Design & Test of Computers*, 17(3):34-42, 2000.
[2] Y. Huang, C.-C. Tsai, N. Mukherjee, O. Samman, W.-T. Cheng, and S. M. Reddy. “Synthesis of scan chains for netlist descriptions at the RT-level” *Journal of electronic testing*, 18(2):189-201, 2002.
[3] R. B. Norwood and E. J. McCluskey. “High-level synthesis for orthogonal scan” *IEEE proceedings VLSI test symposium*, 1997.
[4] M. Hirech, J. Beausang and X. Gu. "A new approach to scan chain reordering using physical design information" *IEEE proceedings ITC*, 1998
[5] D. Berthelot, S. Chaudhuri and H. Savoj. “An efficient linear time algorithm for scan chain optimization and repartitioning” IEEE *proceedings ITC*, 2002.
[6] P. Gupta, A. B. Kahng, and S. Mantik. “Routing Aware Scan Chain Ordering” *IEEE proceedings Asia and Pacific design automation conference*, January 2003.
[7] B. B. Paul, R. Mukhopadhyay and I. S. Gupta. "Genetic algorithm based scan chain optimization and test power reduction using physical information" *IEEE proceedings ITC,* 2006.
[8] Y. Bonhomme, P. Girard, L. Guiller, C. Landrault and S. Pravossoudovitch. “Efficient scan chain design for power minimization during scan testing under routing constraint” *IEEE proceedings ITC*, 2003.
[9] D. Berthelot, S. Chaudhuri and H. Savoj. “An efficient linear time algorithm for scan chain optimization and repartitioning” IEEE *proceedings ITC*, 2002.
[10] K. D. Boese, A. B, Kahng and R. S. Tasy. “Scan chain optimization: heuristic and optimal solutions” internal report CS dept, University of California at Los Angeles, October 1994.
[11] L. Wang, X. Wen and C. Wu. “Design for Testability: VLSI test principles and architectures”, Morgan Kaufmann, 2006.
[12] W. J. Cook, W. H Cunningham, W. R. Pulleyblank, and A. Schrijver. “Combinatorial optimization” *John Wiley & Son, Inc., New York, NY, USA*, 1998.
[13] A.Schrijver, L.Lovasz, B. Korte, H.J. Promel, and R. L. Graham . “Paths, Flows, and VLSI-Layout”
Springer-Verlag, 1990.
[14] M. R. Garey and D. S. Jonhson. “Computers and intractability: a guide the theory of NP-completeness” *Freeman*, 1979.
[15] Lin, C., Lee, M. T., Marek-Sadowska, M., and Chen, K. 1995. Cost-free scan: a low-overhead scan path design methodology. In *Proceedings of the 1995 IEEE/ACM international Conference on Computer-Aided Design*. IEEE Computer Society, Washington, DC, 528-533
[16] www.opencores.org

Responsables de la publication : Nadia Brauner et András Sebő
ISSN : 1298-020X - © Laboratoire G-SCOP