

A global optimization for scan chain insertion at the RT-level

Lilia Zaourar*, Yann Kieffer†, Chouki Aktouf‡

*Laboratoire LIP6/Soc, Université Pierre et Marie Curie, 4 place Jussieu, 75252 Paris Cedex 05, France

Email: lilia.zaourar@lip6.fr

†Laboratoire G-SCOP, Grenoble-INP, 46 avenue Félix Viallet, 38031 Grenoble Cedex 1, France

Email: yann.kieffer@grenoble-inp.fr

‡DeFacTo Technologies, 167 rue de Mayoussard, 38430 Moirans, France

Email: chouki.aktouf@defactotech.com

Abstract—We present a new method for scan chain ordering specifically tailored for RTL-scan and its unique challenges.

Keywords—Design For Test; high-level testing; optimization.

I. INTRODUCTION

A current trend is to raise the level of design methodologies, in order to handle more abstract and more concise structures, increasing designing efficiency. This trend is not always followed in the field of IC testing. Although several proposals have been made to raise full-scan at a higher-level [1] [2] [3], the industry is not ready to take that jump.

Obstacles to raising scan insertion at the RT-level are as follows. First, fault models talk about gate-level entities; so inserting logic to detect such faults at higher-level is counter-intuitive. But it is feasible as demonstrated in [1], [2] and [3]. An example of RTL code modification to include scan is given in Fig. 1 (additional code is presented in boldface).

Second, scan insertion at RTL means that scan logic and functional logic are intermingled and undistinguished by usual synthesis tools: to them, the whole tested designs looks “just functional”. This is a strength of RTL-scan: logic common to both functional and test modes can be optimized away. But this means that no stitching reoptimization can be done during or after synthesis. Whereas gate-level scan takes benefit from low-level information available only late in the flow for scan stitching reoptimization [4], [5], [6], the usual proposal for RTL scan is to do it before logic synthesis. It is a one-step process, independent of what’s before and after; in particular, there is no possibility of improvement through iterations: it has to be done right in just one pass.

Third, scan chain ordering is critical to the successful insertion of scan. A bad ordering can seriously impede placement and routing, increasing congestion and area.

It is to the combination of obstacles 2 and 3—finding a good ordering before logic synthesis, long before any placement information is known—that this paper tries to answer to. We use a graph model for the estimation of the proximity of FFs in the placed design in order to compute a good stitching ordering. Before we present our model, we explain why we use total wirelength as a measure for the quality of scan chain orderings.

```
process(clock)
begin
  if clock'event and clock='1' then
    if scan_en='1' then
      Y <= X;
      Z <= Y;
    else
      Z <= X or Y;
      Y <= X and Y;
    end if;
  end if;
end process;
```

Figure 1: Example scanned VHDL process

II. WIRELENGTH AS PRIME PARAMETER FOR STITCHING ORDERING

Intuitively, a good stitching ordering is such that the added wires between FFs for test mode operation are not too long.

The parameters that are seriously impacted by the stitching ordering are performance (i.e. timing closure), congestion during routing, power consumption in both functional and test modes. We use the total wirelength of the design as a measure of the quality of a stitching ordering.

III. PROXIMITY GRAPH

We have defined our objective as minimizing the total wirelength of the scanned design, through the computation of a good scan chain ordering.

Since no placement information is known at scan-insertion time, one has to anticipate what will happen during the placement and routing steps.

To this end, we introduce the proximity graph (P-graph) model of the design. The P-graph has as vertices the FFs of the design, and edges correspond to (shortest) paths between FFs. Those paths can use wires either in the direction of information transmission or in the reverse direction. They cannot go through another FF. Furthermore, the number of gates traversed is limited to a fixed threshold t . The edges of the graph are labeled with the length of the underlying path, where the length is counted as the number of gates traversed by the path.

Fig. 2 gives an example design together with the corresponding P-graph.

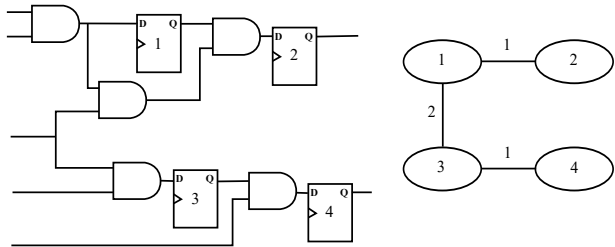


Figure 2: (a) Design example; (b) Corresponding P-graph.

Since there is no canonical mapping from RTL to FFs and gates, a kind of synthesis has to be done to generate a P-graph from RTL code. More details are given in section V. Note that although RTL scan insertion does include a lightweight synthesis step, the RTL scan tool works completely independently from the logic synthesis tool, which comes later in the flow.

IV. STITCHING ORDERING OPTIMIZATION

Although the P-graph is not a complete graph (i.e. some pairs of vertices are not connected), it is possible to reduce the scan chain ordering optimization problem to the Traveling Salesman's Problem. Note that connections in the scan chain can happen outside the P-graph; indeed, the P-graph may very well not contain a Hamilton path.

For scan chain optimization, a classical TSP heuristic (a 2-approximation algorithm) is adapted to our problem.

V. IMPLEMENTATION

Our method for computing scan chain orderings for RTL scan insertion has been implemented in an experimental version of DeFacTo Technologies' HiDFT-SIGNOFF®. The mapping from RTL code to a generic netlist is done by a commercially available software library; P-graph generation and TSP optimization have been implemented using the Boost Graph Library. The threshold value t (defined in section III) is fixed to 4.

VI. NUMERICAL RESULTS

Table I gives actual wirelength values (after placement & routing) for 11 ITC'99 benchmarks and 3 Opencore Verilog designs. Gate-level scan values (column GL-S) are from Magma's Talus integrated flow, as well as logic synthesis, placement and routing. RTL-scan values (column RTL-S) are from our enriched version of HiDFT-SIGNOFF®. The slack column shows the overhead for RTL scan. It is negative except for designs with less than 250 FFs, and when positive, doesn't exceed 6%.

Insertion times are also given in the last two columns. The times are split into two components: P-graph computation and TSP solution. Note that most of the time is spent in establishing the proximity-graph. The longest overall time is for b19 with more than 6000 FFs, at less than 6 minutes.

Table I: Wirelengths and insertion times

Design	#FF	Wirelength		Slack (%)	Insertion time (ms)	
		GL-S	RTL-S		P-graph	TSP
ITC 99 Benchmarks (VHDL)						
b09	28	2.06	1.63	-20.59	<1	<1
b10	17	1.94	2.06	5.97	10	<1
b11	31	4.77	5.03	5.38	20	10
b12	121	12.6	12.92	2.59	200	20
b13	53	3.39	3.32	-2.25	20	<1
b14	245	63.64	64.2	0.96	3s	110
b15	449	126.3	122	-3.39	6s	320
b17	1415	395.6	370.4	-6.37	30s	3s
b18	3320	1187.2	922.6	-22.29	3m	10s
b19	6642	2329.6	1858.2	-20.24	5m	20s
Opencore designs (Verilog)						
Simple-spi	132	11.5	10.4	-8.95	100	20
Biquad	204	34.1	31.3	-8.14	350	80
Ac97	2289	247.7	238.5	-3.71	30s	8s

VII. CONCLUSION

We presented both a metric for the quality of scan chain ordering, and a method to compute good stitching orderings. The method has been implemented and validated on medium-sized designs, showing reasonable computation times, and overall improved wirelengths over Magma's Talus' gate-level scan module.

Perspectives of this work are a more thorough validation of the method; further optimization by taking into account more local factors; combining the method with power-reducing methods for test-mode; and extending the method to the setting of partial scan.

REFERENCES

- [1] S. Roy, "RTL based scan bist," in *Proc. VHDL International User's Forum (VIUF)*, 1997.
- [2] C. Aktouf, H. Fleury, and C. Robach, "Inserting scan at the behavioral level," *IEEE Des. Test. Comput.*, vol. 17, pp. 34–42, 2000.
- [3] Y. Huang, C.-C. Tsai, N. Mukherjee, O. Samman, W.-T. Cheng, and S. M. Reddy, "Synthesis of scan chains for netlist descriptions at RT-level," *J. Electronic Testing*, vol. 18, no. 2, pp. 189–201, 2002.
- [4] M. Hirech, J. Beausang, and X. Gu, "A new approach to scan chain reordering using physical design information," in *IEEE Proc. ITC*, 1998.
- [5] P. Gupta, A. B. Kahng, and S. Mantik, "Routing aware scan chain ordering," in *IEEE Proc. Asian and Pacific Design Automation Conference*, 2003.
- [6] P. Gupta, A. B. Kahng, I. I. Mandoiu, and P. Sharma, "Layout-aware scan chain synthesis for improved path delay fault coverage," in *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, 2005.