Scanning Datapaths : a Fast and Effective Partial Scan Selection Technique

M. L. Flottes, R. Pires, B. Rouzeyre, L. Volpe
Laboratoire d'Informatique, de Robotique et de Micro-électronique de Montpellier, U.M. CNRS 9928
161 rue Ada, 34392 Montpellier Cedex 5, France

1. Introduction

Partial scan DFT is a commonly used technique for improving testability of sequential circuits while maintaining overhead as low as possible. In this context, the selection of the partial scan chain [1] is usually performed at gate-level (e.g [2],[3]). In this paper, we present a method for quickly selecting the partial Scan Chain (SC) in datapath-like circuits. The so-obtained SC is such that the number of scan FFs is optimized and such that the achievable fault coverage is the same than with full scan approach.

2. Partial scan selection at RT level

Basically, our method consists in pre-selecting at RT level a set of registers from which scan FFs will be selected at gate-level. The principle of this method is illustrated in Fig.1. Its main benefit is to avoid to consider all FFs as candidates for scan and thus it saves considerable CPU time. Given the RTL description of the datapath, we first partition the set of registers R into two subsets R_1 and R_2. R_2 contains registers which are totally testable. We can predict at RT level that they will not present any testability problems at gate-level using the testability analysis described in [4]. R_2 registers are eliminated from the candidates for scan. Then, we determine a sufficient subset R_s of R_1 to be candidate for scan. As shown in the following section, the actual SC is built up at gate-level using an ATPG-based approach using R_s instead of R as a starting point.

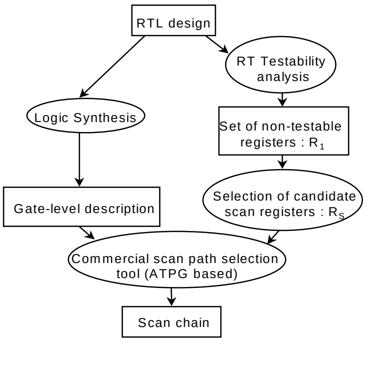
RTL design
RT Testability analysis
Logic Synthesis
Set of non-testable registers : R_1
Gate-level description
Selection of candidate scan registers : R_S
Commercial scan path selection tool (ATPG based)
Scan chain

Figure 1 Scan chain principle

Gate-level information relevant for partial scan issues can be derived from the RT-level testability classification presented in [4]. Following rules are detailed in [5] and are used to make the insertion of registers in the SC :

1. ***Some registers have to be discarded a priori*** :the registers which are already detected as being controllable and observable.

2. ***Some registers have to be selected a priori :*** all registers detected as being non controllable and observable **and** detected such that their own controllability/observability can not be influenced by the addition of other registers in the SC. As a side effect, their inclusion in the SC may solve testability problems of other registers..

3. ***A minimal set of registers have to be selected among remaining non testable registers :*** these registers belongs to loops or reconvergences paths [5]. The selection is done using Algorithm 1 in order to break all these loops and reconvergence paths**.** In spite of the exhaustive nature of this algorithm, the convergence is fast due to the small number of registers under consideration

```
Input : The set F of registers belonging loops and reconverge paths
Output : The subset E of F of registers marked as scan..
For each subset E of F following increasing cardinality
    {every registers belonging to E is transformed in scan registers
    Testability Analysis
    if {F is empty} return(E)}
```

Algorithm 1 Exhaustive algorithm for breaking loops and reconvergence paths

3. Scan Path at gate-level

Let's first recall that not all FFs in a scan register need to be included in the SC. Secondly since RT testability analysis can be pessimistic and since R_S is a by-product of this analysis, not all R_S registers need to be included in the SC. Thus, the actual SC is determined by 1/ expanding the design to gate-level, and 2/ using the gate-level SATPG-based tool from Sunrise [6]. The principle of this tool is the following: first, the fault coverage (fc) of the full scan design is determined. Then, given a set of candidate FFs, generally all of them minus one, the SATPG is run on the modified version of the design in which all these FFs are inserted into the SC. If fc is reached, half of FFs are taken as the new SC and the process is iterated. If not, three quarters of the FFs are incorporated in the SC. And so on and so forth in a dichotomic manner. Thus, with regard to this scan selection technique, the benefit of our approach resides in the use of R_S as the starting set instead of R.

4. Results

This method has been applied to several behavioral synthesis benchmarks. The RTL descriptions are obtained using our high level synthesis tool [7]. All the designs have a constant bitwidth (equal to 8 in the presented experiments). EX1 to EX5 are different versions of the HLS benchmark example borrowed from [8]. EX6, EX7 and EX8 are elliptical filters borrowed from [9]. The RTL designs have been extended to gate-level using Synopsys [10]. In examples EX1 to EX5, the only testability bottlenecks are not due to loops or reconvergence paths thus, the Algorithm 1 for scan pre-selection do not apply. Designs EX6 to EX8 present all kinds of testability problems. Table 1 summarizes comparisons of our partial scan extraction approach with the SATPG-based method. In both strategies, we constrained the gate-level scan-path extraction tool to obtain the fault coverage of full-scan versions of the designs. ATPG runs have been limited to 24 hours. For all examples, additional CPU time required for scan registers pre-selection is lower than one second.

		EX1	***EX2***	***EX3***	***EX4***	***EX5***	***EX6***	***EX7***	***EX8***
Designs information	full scan designs fault coverage	97.14 %	97.77 %	99.53 %	99.52 %	99.53 %	99.23 %	99.23 %	83.64 %
	# registers	5	6	5	5	6	7	7	12
Scan Path Extraction starting with all FFs	# actual scan registers	2	1	2	1	2	6	1	2
	# scan FFs	2	1	2	1	2	19	1	2
	# ATPG runs	6	6	7	6	7	7	7	8
	∑ ATPG CPU time	42.98h	923.11s	110.24s	312.36s	9809s	3 days	90.39s	250730s
Scan Path Extraction starting with Pre-selected FFs	# non-testable registers (R1)	3	1	3	1	2	3	1	12
	# candidate scan registers (R_S)	2	1	2	1	2	2	1	2
	# actual scan registers	2	1	2	1	2	2	1	2
	# scan FFs	2	1	2	1	2	14	1	2
	# ATPG runs	4	3	5	5	5	3	5	5
	∑ ATPG CPU time	11.67h	18.28s	46.94s	51.39s	14906s	1 day	17.52s	133497s
Gains	selected FFs reduction	60%	83%	60%	80%	66%	71%	85%	83%
	ATPG CPU time reduction	72%	98%	57%7777	83%	-51%	66%	80%	46%
	ATPG runs reduction	33%	50%	28%	16%	28%	57%	28%	37%

Table 1 Partial scan strategies comparison

References

[1] V.D. Agrawal, K-T. Cheng, DD. Johnson, T. Lin "Design circuits with partial scan", IEEE Design and Test of Computers, Vol. 5, pp:8-15, April 1988.

[2] V. Chickername, J. Patel: "An optimization based approach to the partial scan design problem", Proc.. ITC 90, pp:377-386.

[3] P.S Parikh, M. Abramovici "A cost-based approach to partial scan". Proc. 30th DAC 1993, PP:255-259.

[4] M.L. Flottes, R. Pires, B. Rouzeyre, "Analysing testability from behavioral to RT level", Proc. European Design and Test Conference" 1997, Paris, pp:158-165.

[5] M.L. Flottes, R. Pires, B. Rouzeyre, L. Volpe, "High Level Synthesis for Partial Scan". Proc. HLDVT'97, Oakland USA, Nov. 14-15, 1997.

[6] Sunrise tests system: reference manual. Version2.3.b, February 1997.

[7] B. Rouzeyre, D. Dupont, G. Sagnes, 'Component Selection, Scheduling and Control schemes for High Level Synthesis', ED&T94, pp482-489, France, 1994.

[8] C.-J. Tseng and D. Siewiorek, "Automated Synthesis of Data Paths in Digital Systems", IEEE Transactions on CAD, vol.5, No. 3, July 1986.

[9] P. Dewilde, E. Deprettere, R. Nouta , "Parallel and pipelined VLSI implementation of signal processing algorithms", In VLSI and Modern Signal Processing. S.Y.Kung et al. Editors. Prentice Hall. pp: 257-260.

[10] Synopsys reference manual : "design analyzer". Version 3.5.a, 1996.