Automatic Scan Insertion and Pattern Generation for Asynchronous Circuits

Aristides Efthymiou
Dep. of Computer Science,
Un. of Manchester,
Manchester M13 9PL, UK
a.efthymiou@cs.man.ac.uk

Christos Sotiriou
Institute of Computer Science,
FORTH
Heraklion, GR 711 10 Greece
sotiriou@ics.forth.gr

Douglas Edwards
Dep. of Computer Science,
Un. of Manchester,
Manchester M13 9PL, UK
d.edwards@cs.man.ac.uk

Abstract

This paper presents 3ϕLSSD, a novel, easily-automatable approach for scan insertion and ATPG of asynchronous circuits. 3ϕLSSD inserts scan latches only into global circuit feedback paths, leaving the local feedback paths of asynchronous state-storing gates intact. By employing a three-phase LSSD clocking scheme and complemented by a novel ATPG method, our approach achieves industrial quality testability with significantly less area overhead testing the same number of faults compared to full-scan LSSD. The effectiveness of our approach is demonstrated on an asynchronous SOC interconnection fabric, where our 3ϕLSSD ATPG tool achieved over 99% test coverage.

1. Introduction

In the class of "bundled-data" asynchronous circuits (data arrive as a bundle w.r.t. a request or acknowledgement signal), the datapath is a collection of pipeline latches and combinational circuits, as in a standard synchronous system. Thus, testing the datapath is straightforward and standard testing techniques can be applied. On the other hand, testing asynchronous control is harder because there is a large number of local state variables which are not updated with a global clock signal. Moreover, C gates, which are commonly used as storage elements, hold the state in a feedback loop, rather than in a conventional latch or flip/flop. C gates are effectively a special form of set-reset latches, e.g. a 2-input C gate waits until both its inputs assume the same logic value, then sets its output to that value.

Conventional, full-scan methods have been successfully employed for testing asynchronous circuits [1]: each feedback loop is broken, in test mode, by inserting a LSSD-type scan-latch. The contribution of this paper is a systematic partial-scan approach, which avoids inserting scan latches to break the *local* feedback loops present in C gates.

1530-1591/04 $20.00 (c) 2004 IEEE

2. Testing C gates

Since the feedback loops in the C-gates are not 'scanned', sequential patterns must be generated to provide adequate fault coverage. In our experience, due to the existence of the feedback path, commercial ATPG tools are unable to generate test patterns that produce satisfactory fault coverage. Even adding gates to control the feedback value, does not raise the coverage of the patterns over 89%.

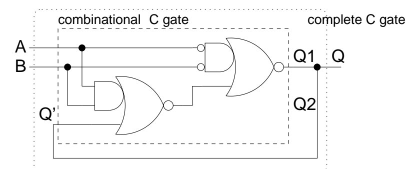

Figure 1. The 2-input C-gate.

Figure 1 shows an implementation of a 2-input C gate using common std. cells. If the feedback is removed, i.e. the output feeding back is considered as another primary input, a C gate is transformed into a combinational circuit for which ATPG can trivially produce test patterns. It can be shown (the proof is not included due to space limitations) that the faults in the "complete C gate" dominate those of the "combinational C gate", thus the patterns generated for the latter can be used for testing the complete C gate.

Care must be taken to put the automatically generated patterns in the right order, so that the output of the C gate, when a pattern is applied, becomes the Q' input of the next pattern. Moreover hazards, which can occur when both inputs of the C gate change in opposite directions, should be avoided. Table 1 shows a safe pattern sequence for a 2-input C gate, which achieves 100% stuck-at fault coverage. Note that two patterns are used twice to avoid hazards (marked with *).

	global					local		
a	1	1	0	0	1	0	0	1
b	1	0	0	1	1	1	0	0
q	1	1	0	0	1	1	0	0
					*		*	

Table 1. Test patterns for the 2-input C gate.

3. Testing asynchronous interconnect

The C gate test patterns presented above can be used to test CHAIN [2], an asynchronous interconnect fabric, using a partial scan approach where scan-latches are inserted only at global feedback loops. All the interconnect building blocks contain variations of the same circuit, a 1-of-5 pipeline latch (fig. 2), which stores and forwards the transmitted data. The global loop indicated in the figure is broken by placing a scan-latch just before the inverter.

Figure 2. 1-of-5 pipeline latches.

The parallel C gates inside the pipeline latches all exhibit the same connectivity: an 'independent' input (a) coming from the upstream pipeline latch, one 'common' input from the scan-latch (b for the acknowledge signal) and their outputs become the 'a-inputs' of the next stage[1]. The intention is to use the same patterns for all five parallel C gates, thus testing the same faults in all C gates simultaneously. This also helps to keep the number of patterns and the test time to a minimum.

Many of the C-gate patterns have the property that the produced gate output value is the same as that of their 'independent' input a. These are the first five patterns in table 1, called 'global'. The importance of these patterns is that they can be used to exercise all the pipeline latches along a link route (from an initiator to a target) simultaneously, since the same values are generated at the a-inputs of all the C gates in all the latches in the link route. This saves a significant amount of test time.

For the remaining C gate patterns, called 'local' here, each pipeline latch must be tested independently. To apply the local patterns to a specific latch, the upstream latches in the link route must be made to generate the appropriate values, while the ones downstream must be set to propagate the results to the end of the link route.

Having the sets of test patterns required for the individual parts of a CHAIN interconnection, it is relatively straightforward to automate the production of a complete sequence of test patterns for a given topology. Such a program has been implemented and tested in a number of CHAIN topologies.

3.1. Evaluation

The presented method was evaluated using an interconnect fabric with 3 initiators and 5 targets. A total of 50 scan latches were required; 98 including the boundary scan-latches at the inputs and outputs. A full-scan approach [1] would require at least 146 scan-latches (194, incl. boundary scan). Thus, our partial scan approach requires about a third of the scan elements (half incl. boundary scan). Although the scan-latches required are larger, the results for the whole interconnect show that 3ϕ LSSD reduced the total area by 60%. Moreover, since for both types of scan latches the path from the input to the output is through a multiplexer and two latches, the delay overhead of both methods is identical.

The 3ϕ LSSD automated test pattern generator produced a sequence of 424 patterns for the interconnect, which achieve a total test coverage of over 99.5%. This coverage includes faults in the boundary scan latches and the scan chain. In fact, the only faults that are undetected are in the control blocks of the arbiters, where a metastability filter, implemented with standard cells, requires the use of OR gates with all inputs tied together, which cannot be tested for stuck-at-0 faults at their inputs individually.

Acknowledgements

This work was funded by the EU through the IST-2002-37796 ASPIDA project. The authors are grateful for this support.

References

[1] K. van Berkel, A. Peeters, and F. te Beest. Adding synchronous and LSSD modes to asynchronous circuits. In *ASYNC02*, pages 146–155, Apr. 2002.

[2] W. J. Bainbridge and S. B. Furber. CHAIN: A delay insensitive CHip area INterconnect. *IEEE Micro*, 22(5):16–23, Sept/Oct 2002.

1 For routing elements of the interconnect, the C gates have 2 'common' inputs; the second comes from a control block which selects one of two parallel pipeline latches and is controlled by a scan-latch.