Partial Scan Test Generation for Asynchronous Circuits Based On Breaking Global Loops

Dilip P.Vasudevan

School of Informatics
University of Edinburgh, UK
Email: D.P.Vasudevan@sms.ed.ac.uk

Aristides Efthymiou

School of Informatics
University of Edinburgh, UK
Email: aefthymi@inf.ed.ac.uk

Abstract: Asynchronous circuits without global clocks are hard to test. Scan design methods provide an effective way to test these circuits. Full scan design seems to be a perfect alternative but at the cost of area overhead. So partial scan based design helps to overcome this area overhead problem and giving a better fault coverage. This paper presents a partial scan based ATPG methodology to generate tests for asynchronous circuits with state holding elements other than latches. The partial scan elements selection is adopted from the conventional Cheng's method based on strongly connected components. One more step of cyclic to acyclic conversion is carried out to generate test patterns for the asynchronous circuits compared to the conventional synchronous circuit test generation. These test patterns are then used to test the original circuit. This method is compared with full scan and other methods with respect to fault coverage and area overhead.

I. Introduction

Asynchronous circuit design involves using designs without global clocks. This feature seems to be more advantageous compared to conventional synchronous design with respect to power dissipation, noise immunity and clocking. Several CAD tools for synthesis of asynchronous circuits are available currently. [1-3]. But it poses serious issues in terms of test generation. Due to the lack of global clock, the observability and controllability of the design is diminished and hence novel test generation methods are required for testing these circuits. Full scan testing has been introduced long back for synchronous circuits. Applying the same technique for asynchronous circuits [14] produces considerably good test coverage but at the cost of huge area overhead. To advance further partial scan test generation provides comparable fault coverage with lower area overhead DFT.

This paper proposes a novel partial scan design methodology and a technique for generating test patterns for asynchronous circuits. Generating test patterns with high stuck-at fault coverage and achieving a lower area overhead compared to existing full scan methods forms the motivation for this work.

Two main contributions are made: an adaptation of the synchronous partial scan method [9] for asynchronous circuits and a cyclic to acyclic circuit conversion which is used both for the partial scan selection procedure and for preparing the circuit for test pattern generation. Fault coverage of 76-96% was obtained by this method while reducing the area overhead compared to full-scan by up to 100% in many cases.

The organization of the paper is as follows. Section 2 gives the background. Section 4 describes the proposed algorithms for the test method. Section 4 briefs the test methodology. The results are analyzed in Section 5 with two working examples. The paper is concluded in section 6.

II. Background

Asynchronous circuits use combinational loops to store state. There are two types of loops, namely global and local loops. *Local loops* are the combinational loops present in the state-holding gates like C-elements or set-reset latches. The familiar flip-flop also contains a local loop, but it is hidden from test tools since a flip-flop is a cell on its own in standard cell libraries and does not pose any problems in testing. *Global loops* are longer loops formed outside these gates and are used for creating asynchronous state machines.

Asynchronous full-scan methods [14] break *all* these loops in test mode using LSSD-type scan latches. This simplifies testing as the circuit becomes purely combinational in test mode. However the area overhead is enormous, hence motivating our work on partial-scan methods.

Strongly Connected Components

A graph represented by G (V,E) , where V forms the set of vertices and E forms the set of edges is said to be a strongly connected graph if there exists a path from each vertex of the graph to every other vertex. Strongly connected components of the graph are its maximal strongly connected subgraphs.

The algorithm of finding the strongly connected components is a linear time O (V+E) for the graph represented as adjacency list [8]. It uses depth first search to find the components of the graph.

Cyclic to acyclic conversion

Cyclic to acyclic conversion of the circuit should be performed for the effective test generation of asynchronous circuits using a synchronous TPG tool. Several methods for generating an acyclic circuit from cyclic circuits have been introduced [4-7]. Unfortunately, these methods are restricted for cyclic circuits without state holding elements and which do not oscillate. Oscillations are predominant in asynchronous cyclic circuits and also state holding elements like C-elements are commonly found in them.

III. Test Methodology

Several steps involved in this test methodology are discussed in this section. Fig. 1 shows the components involved in test generation. As the circuits dealt in this method are asynchronous circuits, the state graph level description of the circuits synthesized using petrify [1]. The synthesized circuits are converted to graph level representation. Blif2Graph generator converts the circuit representation to graph in which nodes represent the gates and edges represent the connection between the gates. Further to apply Cheng's scan selection method [9], the abstract representation of the graph with only memory elements are needed. The abstract level of graph with memory elements as nodes and path between the elements as edges is created.

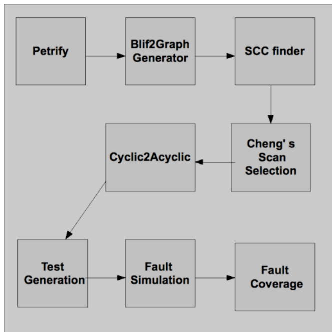

Fig.1. Test Methodology

In the next step, the strongly connected components are identified which aids the Cheng's scan selection algorithm. By applying the Cheng's scan selection method the memory elements to be scanned are selected.

With synchronous designs, the circuit is ready for scan test generation as the global loops are broken, but for asynchronous design, the circuits are still containing the local loops. The C-elements that have not been selected to be scanned constitute these loops. Hence the circuit is passed to cyclic to acyclic converter. Thus, the acyclic partially scanable equivalent of the cyclic partially scanable circuit is obtained. Now the design is passed through a conventional test pattern generator. Synopsys's Tetramax is used for test generation and fault simulation.

IV. Algorithms

The circuit model and the algorithms involved in Global loop breaking, scan selection, cyclic to acyclic conversion are discussed further in detail.

A. Global loop breaking

In [9], the method of global loop breaking involves representing the design into an abstract circuit topology graph. All the vertices in the graph represents the flip-flops in the design and the edges forms the path between the flops comprising of combinational gates and wires. Then the graph is processed to find the strongly connected components [10] present in it, which constitutes to the global loops in the circuit. All the cycles or loops are stored as a list to be processed by the flip-flop selection algorithm. The Breakloop algorithm, outlined below, selects the minimum number of flipflops in the design. Scanning the selected flipflops will cut all the global loops in the design.

```
Repeat
    For every vertex
        Count the frequency of appearance in the cycle list
                Select the most frequently used vertex
                Remove all cycles containing the vertex from the cycle list
  Until cycle list is empty
```

Thus the flipflops selected will form the scan elements in case of synchronous design. In case of asynchronous circuit, the C-elements present in the circuit are also considered as local loop or memory elements. Thus C-elements are added as vertices during the graph representation, before applying the scan selection algorithm. After applying the scan selection algorithm, the selected C-elements and latches will form the scan elements for the design

B. Cyclic to Acyclic Conversion:

Once the feedback wires have been identified, for the purpose of test pattern generation, the circuit must be converted to an acyclic one by replicating the appropriate parts of the circuit. This is similar to the time frame unrolling method used in sequential pattern generation.

The conversion algorithm requires a user specified number of cycle copies. The number of copies will also depend upon the number of cycles present in the original cyclic circuit and whether they are nested or intersected. As the full algorithm is too long to fit in the space available, the interested reader is referred to [11].

V. Working Examples and Results

The overall methodology is explained further by showing the flow through two example circuits namely the majority gate based C-element and benchmark ram-read-sbuf [15]

A. Benchmark 1

A majority gate based C-element is shown in fig 2. The circuit is cyclic and consists of four gates and two local feedback loops. Since the C-element does not have memory elements, the scan selection algorithm does not select any scan element. This example is provided to show the cyclic to acyclic conversion in absence of a memory element. Thus if no memory element is present and still the circuit has loops, the cyclic to acyclic converter will produce an equivalent acyclic circuit and the test patterns generated for this circuit are used to fault simulate the original circuit as shown in fig 2. The acyclic circuit consists of 3 inputs, 1 output and 11 gates. The converted acyclic circuit is fed to the Synopsys's Tetramax to obtain the test patterns. The test patterns obtained are 111, 000, 100, 010, 111, 101, 011 for the pins a, b, c respectively, with c being the initialization pin. The actual patterns used to test the real cyclic circuit are therefore the first two bits of the above sequence. Tetramax was also used for fault simulation and the fault coverage is 88.23%.

B. Benchmark 2

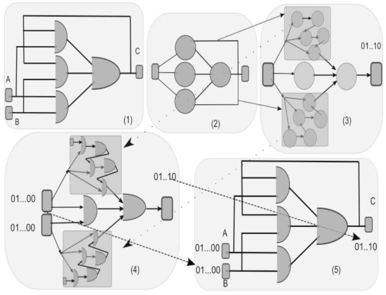

Fig: 2 Test generation C-element

The benchmark ramreadsbuf is shown in fig 3. The circuit is cyclic and consists of 14 combinational gates, 2 C-elements constituting 2 local loops and 1 global loop (highlighted in fig 3). The graph equivalent of the circuit with memory elements as vertices is shown in fig 3.a. Thus the 2 C-elements in the circuit contribute to the two vertices in the graph (fig 3.a). One global loop is present in the circuit, which is shown as 1 feedback loop in the graph (fig 3.a). The gates and memory elements constituting the loop is shown in fig 3.b. One strongly connected component is identified forming a cycle list with 1 cycle.

By applying scan selection algorithm, the C-element c1 will be selected. C1 will be selected at the first pass, as it constitutes to the cycle 1 emptying the cycle list to halt the algorithm. As explained earlier, the circuit has 1 C-element left without being scanned and hence it constitute to the local loop. So the circuit is fed to the cyclic to acyclic converter. The resulting circuit will be a partially scanned circuit free from local loops. The acyclic circuit is fed to the Tetramax to generate the test patterns. These test patterns are then fault simulated over the original partially scanned DUT to obtain the fault coverage. The fault coverage for this benchmark is 96.34%.

C. Results and analysis:

The proposed method was applied to several asynchronous circuits commonly used as benchmarks. The number of faults and the fault coverage results are shown in Table I together with the results from a previous work [12]. We also include the results used as a base for comparison in [12] which use a more conservative hazard detection method proposed in [13].

The fault coverage for the proposed method was found to be between 76-96%. The fault coverage for the Eichelberger's method[13]was between 21.4% and 100%. spin-sim [12] provided a fault coverage between 85.7% - 100%.

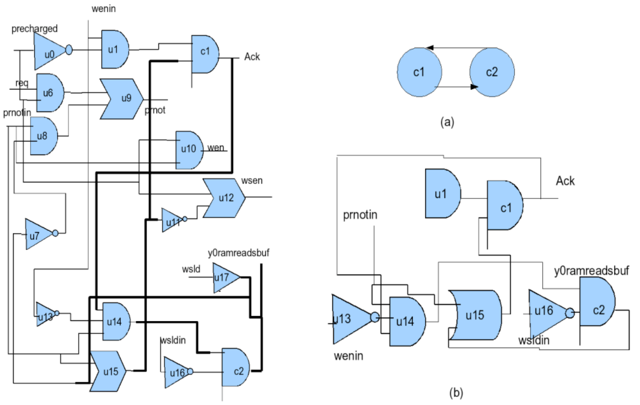

Fig. 3 Global loop in the benchmark ramreadsbuf

A comparison was also made to simply presenting the original cyclic circuit to a state of the art TPG tool (Tetramax) and to full-scan [14]. The results are shown in Table II.

As expected, a standard TPG tool is unable to handle feedback loops and thus finds patterns for less than 50% of the faults in most cases.

The fault coverage for the full scan design ranged from 85.71 – 95.18%. DFT circuit from this method however, had much lower DFT area overhead. In a number of benchmarks scan elements were not required at all (100% reduction), while at the very least this method required half the number of scan elements than full-scan.

It is clear that this method does not achieve as high a fault coverage as desired. The undetected faults in all the circuits tested were found at the internal nodes of C-elements. Even though most faults are excited by the tests produced by this algorithm, not all the faults could be propagated to a primary output or a scan element. Fault simulation reported those faults as not observable. We believe that this is due to the fact that this method currently does not propagate faults sequentially to a primary output. In other words, after a fault is excited, it does not apply further input values just to propagate the fault.

VI. Conclusion

A partial scan design and test pattern generation for asynchronous circuits based on strongly connected components and cyclic to acyclic conversion was introduced in this paper. The selection of the state elements that will be "scanned" is made using an adaptation of the Cheng's method [9]. Test coverage was improved compared to test generated from original latch free circuit without applying DFT.

The proposed method has been applied to a number of benchmarks achieving 76-96% stuck-at fault coverage while reducing the area overhead compared to full-scan by 50 - 100% .

TABLE I
Result – Fault Coverage Comparison

Benchmark	**No of faults**	**Fault coverage**		
		[13]	**[12]**	**Proposed**
C-element	26	-	-	88.23
Half	22	40.0	100	100
Hazard	48	87.9	97	76.19
dff	40	21.4	85.7	76.67
rcv-setup	25	100	100	77.78
chu150	56	97.1	97.1	89.19
chu133	54	96.9	96.9	91.23
mp-forw-pack	60	100	100	92.77
nak-pa	82	100	100	76.78
ram-read-buf	90	100	100	96.34
rpdft	62	100	100	76.78
sbuf-ram-write	110	100	100	96.25
sbuf-send-ctrl	94	59.3	94.9	90.36

TABLE II
Result – Fault Coverage Comparison

Benchmark	Latch Free FC (%)	Full Scan FC (%)	Prop (%)	Loops	Overhead Reduced%
chu133	45.56	85.71	91.23	1 loc	100
chu150	23.33	90.15	89.19	2 loc	50
mp-forward-packet	69.57	95.18	92.77	1g, 1loc	100
ram-read-buf	40.71	93.29	96.34	1g,2 loc	100
sbuf-ram-writ e	22.79	95.00	96.25	2 loc	50
Sbu-send-ctrl	16.20	94.16	90.36	3 local	66.66

References

[1] Cortadella, J.; Kishinevsky, M.; Kondratyev, A.; Lavagno, L.; Yakovlev, A. *Petrify: a tool for manipulating concurrent specifications and synthesis of asynchronous controllers*, IEICE Trans. Inf. and Syst., pages 315-325. March 1997.

[2] K.van Berkel, J. Kessels, M. Roncken, R.Saeijs, and F.Schalij. The VLSI-programming language Tangram and its translation into handshake circuits. *In Proc. European Conference on Design Automation*, pages 384-389, 1991.

[3] A. Bardsley and D.Edwards. Compiling the language Balsa to delay-nsensitive hardware. *In C.D.Kloos and E.cerny, editors, Hardware Description Languages and their Applications{CHDL}*, pages 89-91, Apr. 1997.

[4] Stephen A.Edwards, Making cyclic circuits acyclic,Proceedings of the 40th conference of Design Automation,2003,159-162,Anaheim,CA,USA

[5] .Sharad Malik. *Analysis of cyclic combinational circuits. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 13(7):950--956, July 1994.

[6] Herbert Weinblatt, *A New Search Algorithm for Finding the Simple Cycles of a Finite Directed Graph,*J. ACM}, volume-19,1,1972,43-56, ACM Press,New York, NY, USA.

[7] Gabriel Nivasch, *Cycle detection using a stack*,Inf. Process. Lett.,90,3, 2004,135—140, Elsevier North-Holland, Inc., Amsterdam, The Netherlands.

[8] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms, Second Edition*. MIT Press and McGraw-Hill, 2001. Section 22.5, pp.552–557.

[9] K.T. Cheng and V.D. Agrawal, *A Partial Scan Method for Sequential Circuits with Feedback,* IEEE Transactions on Computers, Vol. 39, pp. 544–548, April 1990.

[10] R. V. Hudli and S. C. Seth, "*Testabitity analysis of synchronous sequential circuits based on structure data*," in Proc. Int. Test Conf., Aug. 1989, pp. 364-372.

[11] D.P.Vasudevan and A.Efthymiou, Partial Scan Test Generation for Asynchronous Circuits, Tech-Report, ICSA,Univeristy of Edinburgh.

[12] F. Shi and Y. Makris, "*Spin-sim: Logic and fault simulation for speed-independent circuits:*' in ***Pmceedings of lnternationall*** *Test* ***Conference,2004,597-606.***

[13] E. B. Eichelberger. Hazard detection in combinational and sequential switching circuits. IBM Journal of Research and Development, 9:90--99, March 1965.

[14] e Beest, F., Peeters, A., Van Berkel, K., and Kerkhoff, H.. *Synchronous Full-Scan for Asynchronous Handshake Circuits*. J. Electron. Test. 19, 4 (Aug. 2003), 397-406.

[15] www.async.ece.utah.edu/ATACS/