TECHNIQUES FOR AT-SPEED TESTING OF VLSI ASIC DESIGNS

Sujit Dey **Miodrag Potkonjak**
C&C Research Laboratories, NEC USA
Princeton, NJ 08540

Abstract - This paper presents ***non-scan*** **design-for-testability techniques applicable to register-transfer (RT) level data path circuits, which are usually very hard-to-test due to the presence of complex loop structures. We develop a new DFT measure, and utilize the RT-level structure of the data path together with adding constants, for cost-effective re-design of the circuit to make it easily testable, without having to either scan any FF, or break loops directly. The non-scan DFT technique was applied to several data paths. Experimental results demonstrate the feasibility of producing non-scan testable data paths, which can be tested at-speed, with marginal area overheads.**

1 Introduction

With the rapid increase in the complexity of DSP ASIC designs, and ever shorter time-to-market requirements, testing plays an increasingly important role in the product cycle. Recently, several high level synthesis approaches have been proposed to generate easily testable data paths for both Built-In-Self-Test (BIST)-based testing methodology [1, 2], and Automatic Test Patern Generation (ATPG) methods [3]. However, almost all BIST-based approaches assume a scan design methodology since random testing is not well-suited for sequential circuits. Also, almost all the ATPG-based high level synthesis for testability approaches, assume the use of scan registers to make the data paths testable.

The scan-based techniques have the disadvantage that the test application time is very large compared to non-scan designs, since the test vectors have to be shifted through the scan chain. On the other hand, non-scan DFT techniques do not require to scan any FFs, thus eliminating the need to shift test vectors through scan chains, and greatly reducing the test aplication time.

However, the biggest disadvantage of scan-based DFT techniques is that the test vectors cannot be applied at the operational speed of the circuit, that is, test vectors cannot be applied at consecutive clock cycles. The inability of scan designs to be tested at-speed assume significance in light of recent studies, which show that a stuck-at test set applied *at-speed* identifies more defective chips than a test set having the same fault coverage but applied at a lower speed [4, 5]. The studies motivated researchers to investigate gate-level non-scan DFT techniques to make sequential circuits testable by introducing

0-7803-2123-5/94 $4.00 © 1994 IEEE

controllability and observability points [5]. The main advantage of the non-scan designs is that the test vectors can be applied at-speed.

The dependencies of the flip-flops (FFs) of a sequential circuit is captured by an S-graph [6]. It has been empirically shown that the presence of loops in a sequential circuit is the main source of problem for sequential ATPG. An effective gate-level partial scan approach selects scan FFs in the minimum feedback vertex set (MFVS) of the S-graph, so that all loops, except self-loops, are broken, and the sequential depth is minimal [6, 7].

1.1 The New Approach

This paper presents new DFT techniques to make data paths testable, without using scan registers. We use the Register Transfer level structure of data paths to introduce the EXU S-graph, which captures the dependencies between the Execution Units (EXUs) of the data path. We show that the choice of EXUs (their outputs) as the nodes to be made controllable/observable is more effective than the choice of FFs (registers at the RT-level) used by traditional scan and non-scan DFT techniques [6, 5], since the MFVS of the EXU S-graph is a lower bound to the MFVS of the S-graph of its registers. Also, as opposed to making the same node controllable and observable (as in scan approaches), we use a more cost-effective distributed approach, where some nodes are made controllable, while some other nodes observable.

This paper shows that breaking all the loops is not necessary to make a circuit testable. We develop a new testability measure, based on k-level controllability and observability of loops. It is demonstrated that it suffices to make loops k-level controllable/observable, instead of directly breaking them, to make the data paths highly testable. We utilize the RT-level structure of the data path, and the k-level measure, for cost-effective re-design of the circuit to make it easily testable, without having to scan FFs, or break loops directly.

The advantages of the DFT approach at the RT-level are numerous. As shown in the paper, the complexity of the EXU S-graph is significantly lower than the FF S-graph. Moreover, knowledge of the RT-level structure as well as the functions of the RT-level components are utilized to develop new cost-effective non-scan DFT techniques. Controllability/observability points can be implemented using register files and constants, and large cliques of size k of FFs can be broken by choosing a single control/observe point, instead of having to control/observe $k - 1$ FFs.

We applied our technique on several moderately sized data paths. The experimental results demonstrate the effectiveness of the k-level heuristic, and our non-scan DFT technique to design highly testable data paths, with nominal hardware overhead. Besides the main advantage of at-speed testing, experimental results also demonstrate that the hardware overhead and the test application time required for the non-scan designs is significantly lower than the partial scan designs.

2 Scan and Non-Scan DFT of RT-level Data Paths: An Illustration

Figure 1(a) shows the register-transfer (RT) level data path for 4th order IIR cascade filter, synthesized from behavioral description using the HYPER high-level synthesis system [8]. The data path has several loops involving the registers. As can be expected, sequential ATPG is very difficult for the data path, as indicated in Table 2 by the row *Orig.*

The testability of the data path can be improved using partial scan techniques to break all the loops of the circuit. Since the MFVS of the S-graph of the data path is 3, breaking all the loops needs scanning at least 3 registers, namely LA1, LA2, and LM1. For the 20-bit IIR filter data path shown in Figure 1(a), 60 scan FFs are needed by the gate-level partial scan tool OPUS [9], and Lee-Reddy's partial scan tool [7], shown by the rows *Opus* and *LR* respectively in Table 2. The sequential ATPG program HITEC [10], can achieve 100% test efficiency on the scan designs, requiring 156 test vectors for the Opus design. Besides the high area overhead, the scan designs have high test application time, indicated by the column **Tappl** in Table 2. For instance, the Opus design needs 156*(60+1) = 9516 clock cycles to apply all the 156 test vectors. Most importantly, the scan designs cannot be tested at-speed.

2.1 EXU S-Graph

Unlike a partial scan DFT technique, it is not necessary for a non-scan DFT technique to restrict to only registers the choice of nodes to break, that is, make controllable/observable. Also, as opposed to making the same point controllable as well as observable, it may be more cost-effective to make some points controllable, while some other points observable. In this section, we introduce the *EXU S-graph*, to show that in a data path, (the outputs of) EXUs are better choices for controllable/observable points than registers. Each node in the EXU S-graph represents an EXU in the data path. There exists a directed edge from node u to node v, labelled i, and denoted $u \xrightarrow{i} v$, if there exists a direct path from EXU u to the ith register file of EXU v, without going through any other register.

The EXU S-graph for the data path in Figure 1(a) is shown in Figure 1(b). The EXU S-graph has several loops. There are two loops in the EXU S-graph between M1, A2 and TU1, namely $M1 \xrightarrow{1} A2 \xrightarrow{1} TU1 \xrightarrow{1} M1$, and $M1 \xrightarrow{2} A2 \xrightarrow{1} TU1 \xrightarrow{1} M1$. However, all the loops in the EXU S-graph pass through the two EXUs, A1 and A2. Hence, the MFVS of the EXU S-graph is 2, as opposed to an MFVS of 3 for the Register S-graph.

2.2 Non-Scan DFT of the IIR Cascade Filter

Next, we proceed with the task of non-scan DFT of the data path in Figure 1(a). Since A1 and A2 form the MFVS of the EXU S-graph, making the outputs of A1 and A2 controllable/observable would break all the loops directly,

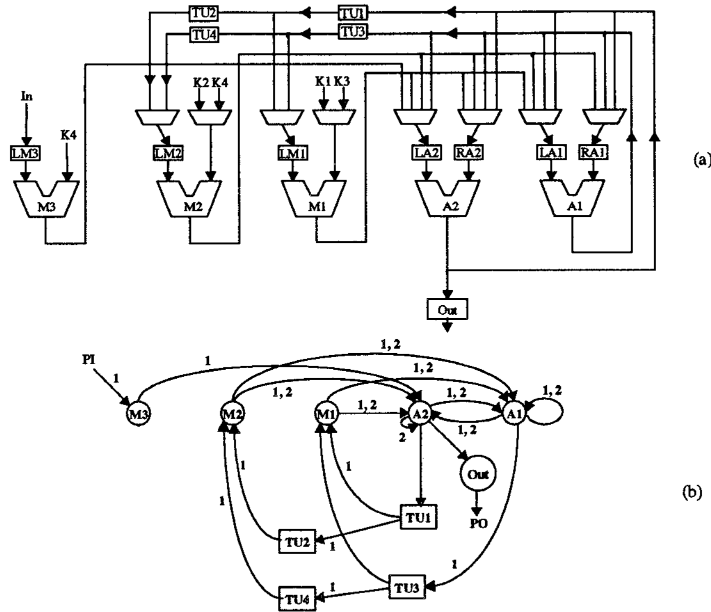

Figure 1: Characteristics of a design implementing the 4th order IIR cascade filter: (a)the RT-level data path, (b) Register S-graph, (c) EXU S-graph

that is, make all the loops *0-level* controllable/observable. That is, any value at the outputs of A1, A2 can be controlled and observed in 1 clock cycle (time frame). Compared to the Register S-graph solution, which requires making three registers controllable/observable, this solution seems better. However, we show that much less expensive non-scan DFT techniques would suffice to make the data path testable.

The EXU S-graph in Figure 1(b) reveals that all loops through A2 are observable, since A2 goes directly to the PO *Out*. Hence, we need to add only a controllability point to output of A2, while adding both a controllability and observability point to the output of A1. Figure 2(a) shows the modified data path of Figure 1(a), with test hardware added (shown in bold) to insert one controllability point at the output of A1 and A2, and one observability point from the output of A2. A test efficiency of 100% could be achieved on the resultant data path, as evidenced by the row *0-lev* in Table 2. The test hardware overhead required for the modified data path is 429 cells, (5.7% of the original data path), which is less than the overhead of 665 cells needed for the scan designs (rows *Opus*, *LR* in Table 2). Besides having the main adavantage of at-speed testing, The number of clock cycles required for test application (column *Tappl*) for the non-scan design is much less than the scan design. However, the main advantage of the non-scan design shown in Figure 2(a) over the scan designs is the ability of at-speed testing.

It is not necessary to make the loops of the data path directly (0-level)

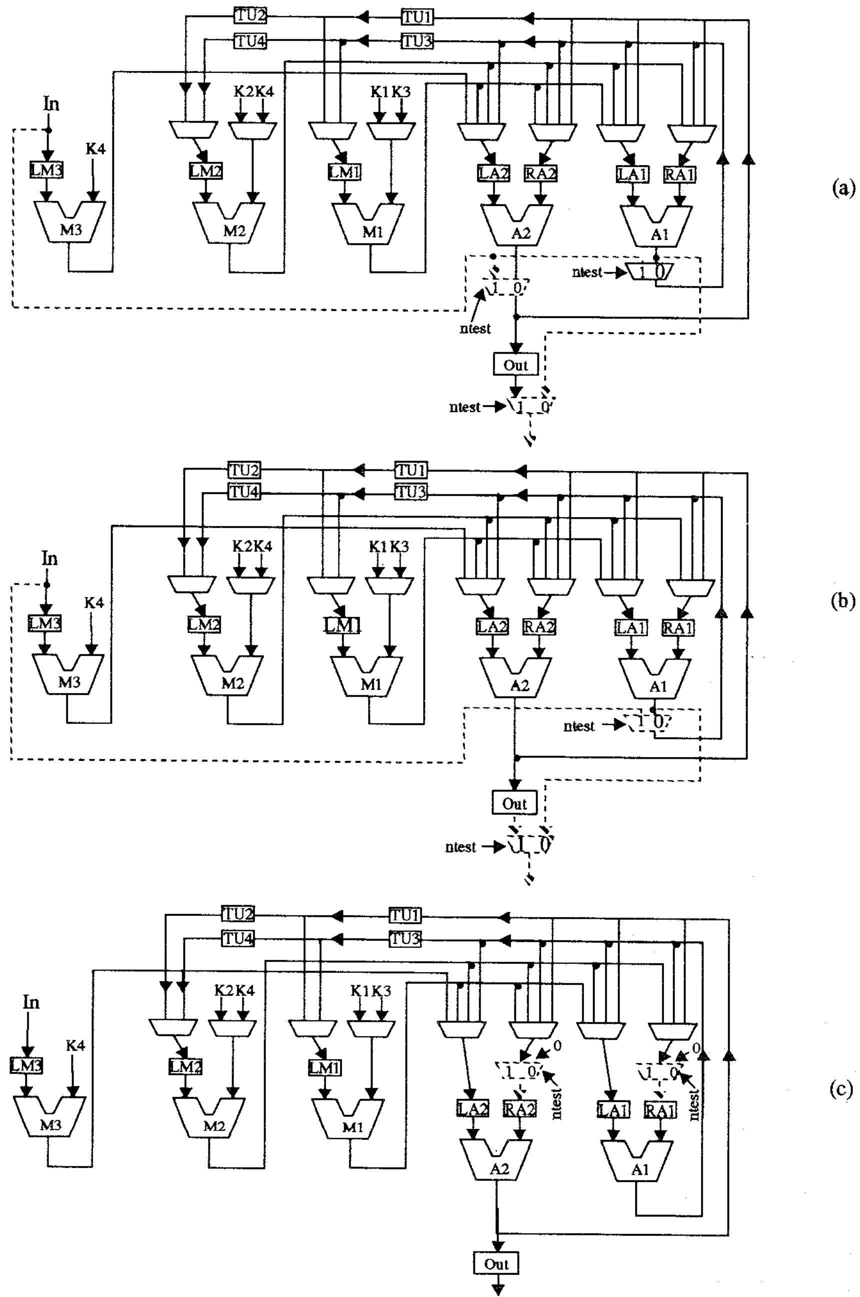

Figure 2: **Non-scan** design-for-testability of the data path in Figure 1(a), (a) 0-level testable data path: all loops are 0-level controllable/observable, (b) 1-level testable data path, (c) 2-level testable data path

controllable/observable. Figure 2(b) shows an alternate testable design, with the non-scan test hardware shown in bold. Instead of adding a controllability point to the output of A2, only a constant ("0", the identity element of the adder) is added to the right register file (RA2) of A2. Any value at the output of A2 can still be justified by at most two time frames. For example, if a value of 9 needs to be justified at the output of A2, in one time frame the registers LA2 and RA2 can be set to appropriate values 9 and 0, and in the next time frame the values of LA2 and RA2 can be justified by *In* and the constant. Adding the constant requires much less hardware overhead than adding a controllability point at the output of A2, since the mux logic associated with the constant signals can be pruned. Note that the new non-scan design makes use of the *register-file based* scheme, to be described later in section 3. The loops through A2 are now 1-level controllable. The resultant (1-level controllable/observable) data path shown in Figure 2(b) has less hardware overhead than the 0-level solution shown in Figure 2(a). Also, a high test efficiency of 98% could be achieved on the resultant data path, as evidenced by the row *1-lev* in Table 2.

The data path in Figure 2(c) demonstrates more effectively the benefits of non-scan DFT at the RT-level, and the new notion of k-level controllable/observable loops. The data path shows the addition of just two constants to the right registers, RA1 and RA2, of the EXUs A1 and A2 respectively. As will be explained in section 3, all the loops in the EXU S-graph now become 2-level or less controllable/observable. The test hardware required is significantly less than the 0-level and 1-level testable data paths shown in Figures 2(a) and 2(b) respectively, as shown in row *3-lev* in Table 2. The area overhead is only 120 cells, as compared to an overhead of 665 cells for the scan design, 429 cells for the 0-level non-scan design, and 349 cells for the 1-level non-scan design. The 2-level testable design, however, has a very high test efficiency of 98%, comparable with the test efficiency achieved by the more expensive scan designs, and the 0-level and 1-level non-scan designs.

3 K-Level Controllable/Observable Loops: A Cost-Effective DFT Approach

In this section, we first define k-level controllable/observable nodes. As mentioned in the previous section, we exclusively consider EXUs (their output bus) as the possible nodes. We introduce non-scan DFT techniques to make nodes k-level conrollable/observable. Next, we define k-level controllable/observable loops in terms of k-level controllable/observable nodes.

Definition 1 *An EXU M is k-level controllable/observable if any value on the output of M can be justified/propagated in at most k+1 clock cycles (time frames). Alternatively, for any value that needs to be justified at the output of M, there exists at least one vector sequence of length at most k+1 that justifies the value.*

The output of an EXU, Z, can be made k-level controllable/observable either by a direct scheme, or a register-file based scheme. In the direct scheme, the EXU output is directly muxed with a k-level controllable node to make Z k-level controllable. The EXU output is made k-level observable by directly muxing it with another node which is k-level observable. Consider the EXU shown in Figure 3(a). Figure 3(b) shows how ALU1 is made k-level controllable and observable using the direct scheme.

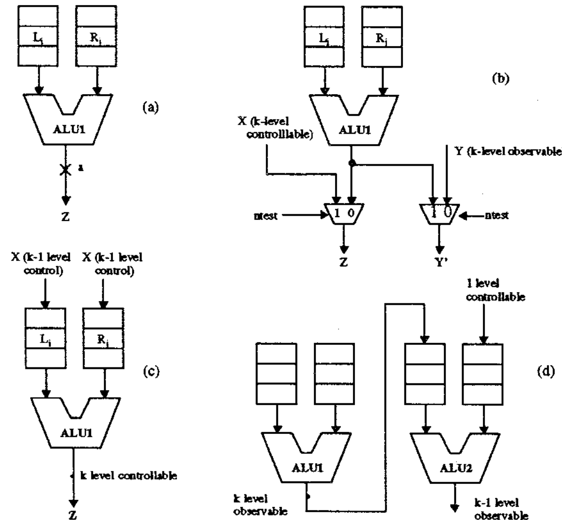

Figure 3: k-level controllability/observability: (a) an EXU, (b) direct scheme for making ALU1 k-level controllable/observable (c) register-file based scheme for making ALU1 k-level controllable, (d) register-file based scheme for making ALU1 k-level observable

In the register-file based scheme, an EXU (output) is k-level controllable if at least one register of each register file of the EXU has a k-1 level controllable input, as shown in Figure 3(c). An EXU is k-level observable if it has an interconnect to a a register file of another EXU, which is k-1 level observable, and whose other register file has a 1-level controllable input. Figure 3(d) shows how ALU1 is made k-level observable.

Definition 2 *A loop is k-level controllable if there is at least one node in the loop which is k-level controllable. A loop is k-level observable if there is at least one node in the loop which is k-level observable.*

Definition 3 *A data path is k-level testable if all loops in the data path are k-level or less controllable and observable.*

Consider the data path shown in Figure 2(c). All loops going through A1 are 2-level controllable and 2-level observable. Similarly, all loops going through

A2 are 1-level controllable/observable. Hence, the data path shown in Figure 2(c) is 2-level testable.

4 Experimental Results

We have developed algorithms which add the minimal hardware possible to make all loops in the data path k-level controllable and k-level observable, for a user-specified value of k, using the non-scan DFT techniques discussed in the previous sections. We applied the non-scan DFT algorithms on different types of data path circuits, synthesized using the high level synthesis system HYPER [8] from behavioral descriptions. In this section, we report results obtained on the following data paths: (1) 4th order IIR cascade filter (4IIRcas), (2) Speech filter (Speech), (3) 5th order elliptical wave digital filter, synthesized using high hardware sharing (EWFhigh).

Table 1 shows various parameters of the data paths obtained using HYPER [8]. The word size of the designs (**Bits**), the number of adders (**Add**), multipliers (**Mult**), registers (**Reg**), multiplexers (**Mux**), and interconnects (**Inter**) are reported. Also, the number of cells needed for the final technology mapped circuit, using the SIS technology mapper from UC Berkeley, is reported in the column **Area**.

Design	Bits	Add	Mult	Reg	Mux	Inter	Area (Cells)
4IIRcas	20	2	3	12	12	20	7486
Speech	20	2	3	12	9	9	6022
EWFhigh	20	1	1	18	23	6	7538

Table 1: Characteristics of the RT-Level Data Paths

The results of applying partial scan and non-scan DFT techniques on the data paths in Table 1 are reported in the Tables 2, 3, 4. The row *Orig* shows the original design. The rows *Opus* and *LR* show the partial scan designs obtained by using OPUS [9] from Univ of Illinois, and Lee-Reddy's tool LR [7]. The designs produced by the non-scan DFT techniques are presented in the subsequent rows, with the rows *2-lev*, *1-lev*, *0-lev*, etc. representing the 2-level, 1-level, and 0-level testable designs respectively.

For each of the designs, column **Test Hardware** summarizes the test hardware that had to be added to the original design to make the circuit testable. In the case of partial scan designs, the number of scan FFs needed is reported. In the case of each non-scan design, the number of constants *c*, the number of control points *cp*, the number of observability points *op*, or the number of dual points *dp* that had to be added to the original design is shown. For each scan and non-scan design, the test hardware overhead, in terms of extra cells used, is shown in the column **Cost**.

We ran the sequential test pattern generator HITEC [10] on the original design, as well as the scan and non-scan designs. The following tables show

the results of running HITEC on each of the designs. The total number of faults and the faults aborted (**Abt**) are shown. Column **TE%** gives the percentage test efficiency, which is the percentage of faults either for which a test could be found or which could be identified as redundant (that is, the percentage of faults not aborted). Also, the number of test vectors needed (**Vec**), and the test generation time taken (**Tgen**) in CPU secs on a Sparc2, are reported. Lastly, the test application time, that is the number of clock cycles needed to apply the test vectors to the designs, is shown in the column **Tappl**.

Type	Test Hardware	Cost (Cells)	Faults		TE%	Vec	Tgen (secs)	Tappl (cycles)
			Total	Abt				
Orig	None	0	10004	9667	3.00	-	23884	-NA-
Partial-Scan DFT								
Opus	60 scan FFs	665	10004	3	99.97	156	226	9360
LR	60 scan FFs	665	10004	3	99.97	168	210	10080
Non-Scan DFT								
2-lev	2c	120	10086	239	97.63	109	6982	109
1-lev	1c,1cp,1op	349	10334	70	99.32	149	686	149
0-lev	2cp,1op	429	10448	5	99.95	268	292	268

Table 2: 4IIRcas: Cost and Effect of several DFT schemes

Type	Test Hardware	Cost (Cells)	Faults		TE%	Vec	Tgen (secs)	Tappl (cycles)
			Total	Abt				
Orig	None	0	8514	8186	4.00	-	19614	-NA-
Partial-Scan DFT								
Opus	20 scan FFs	225	8514	29	99.65	111	193	2220
LR	20 scan FFs	225	8514	16	99.81	131	160	2620
Non-Scan DFT								
1-lev	1c	57	8553	300	96.49	70	7491	70
0-lev	1cp	142	8676	10	99.88	149	412	149

Table 3: Speech Filter: Cost and Effect of several DFT schemes

Type	Test Hardware	Cost (Cells)	Faults		TE%	Vec	Tgen (secs)	Tappl (cycles)
			Total	Abt				
Orig	None	0	9375	9158	2.00	-	29220	-NA-
Partial-Scan DFT								
Opus	280 scan FFs	3085	9375	2	99.97	164	240	45920
LR	280 scan FFs	3085	9375	3	99.96	192	251	53760
Non-Scan DFT								
1-lev	1c	20	9397	271	97.11	218	2625	218
1-lev	1c,1op	161	9519	130	98.63	297	992	297
0-lev	1cp,1op	282	9657	20	99.79	347	250	347

Table 4: EWF with high hardware sharing (EWFhigh): Cost and Effect of several DFT schemes

The Tables 2, 3, 4 show the ability of the non-scan DFT techniques to make the highly untestable data paths very easily testable, with a significantly

smaller test hardware overhead than the scan designs. Consider the results for the data path circuit EWFhigh, shown in Table 4. The original data path has a very low test efficiency (2%). The scan designs achieve very high test efficiency, but the test hardware overhead is very high (3085 cells), and the test application time needed is 45920 clock cycles. On the other hand, the non-scan DFT solutions produced by the proposed techniques achieve comparable test efficiency, with significantly lower area overheads and test application times. For example, the 1-level testable data path uses only 20 extra cells while still achieving a very high test efficiency of 97%. The 0-level non-scan solution achieves almost a 100% test efficiency, while requiring only 282 extra cells, as compared to 3085 required by the scan designs. As expected, the non-scan solutions need significantly lower test application time than the scan designs. Only 218 clock cycles are needed to apply the test vectors to the 0-level design as comapared to 45920 clock cycles needed for the scan implementation.

The experimental results also validate the effectiveness of the k-level controllable/observable loops measure introduced in this paper. The results show that it is not needed to make all the loops directly (0-level) controllable/observable to achieve high test efficiency, as evidenced by the very high test efficiency reported for the k-level testable data paths, $k > 0$. Most significantly, the experimental results demonstrate the feasibility of producing non-scan testable data paths, which can be tested at-speed, with marginal area overheads.
Acknowledgments: We wish to thank Vijay Gangaram for his help.

References

[1] S. S. K. Chiu and C. Papachristou. A Built-In Self-Testing Approach For Minimizing Hardware Overhead. In *Proc of the Intl Conf on Computer Design*, 1991.

[2] L. Avra and E. McCluskey. Synthesizing for Scan Dependence in Built-In Self-Testable Designs. In *Proc. of the International Test Conference*, pages 734 – 743, Oct. 1993.

[3] S. Dey, M. Potkonjak, and R. Roy. Exploiting Hardware Sharing in High Level Synthesis for Partial Scan Optimization. In *Proceedings of the International Conference on Computer-Aided Design*, pages 20 – 25, November 1993.

[4] P.C. Maxwell, R.C. Aitken, V. Johansen, and I. Chiang. The Effect of Different Test Sets On Quality Level Prediction: When is 80% Better Than 90%? In *Proc. of the Intl Test Conference*, pages 358 – 364, Oct 1991.

[5] V. Chickermane, E.M. Rudnick, P. Banerjee, and J. H. Patel. Non-Scan Design-for-Testability Techniques for Sequential Circuits. In *Proc. Design Automation Conf.*, pages 236 – 241, June 1993.

[6] K.T. Cheng and V.D. Agrawal. A Partial Scan Method for Sequential Circuits with Feedback. *IEEE Transactions on Computers*, 39(4):544 – 548, April 1990.

[7] D.H. Lee and S.M. Reddy. On Determining Scan Flip-Flops in Partial-Scan Designs. In *Proceedings of the International Conference on Computer-Aided Design*, pages 322 – 325, November 1990.

[8] J. Rabaey, C. Chu, P. Hoang, and M. Potkonjak. Fast Prototyping of Data Path Intensive Architectures. *IEEE Design and Test*, pages 40 – 51, 1991.

[9] V. Chickermane and J. H. Patel. An Optimization Based Approach to the Partial Scan Design Problem. In *Proc of the Intl Test Conf*, pages 377 – 386, 1990.

[10] T. M. Niermann and J. H. Patel . HITEC: A Test Generation Package for Sequential Circuits. In *Proc. EDAC*, pages 214–218, 1991.