California Scan Architecture for High Quality and Low Power Testing

Kyoung Youn Cho, Subhasish Mitra, and Edward J. McCluskey
Center for Reliable Computing (CRC)
Department of Electrical Engineering
Stanford University, Stanford, CA
{kycho, smitra, ejm}@crc.stanford.edu

Abstract

This paper presents a scan architecture—California scan—that achieves high quality and low power testing by modifying test patterns in the test application process. The architecture is feasible because most of the bits in the test patterns generated by ATPG tools are don't-care bits. Scan shift-in patterns have their don't-care bits assigned using the repeat-fill technique, reducing switching activity during the scan shift-in operation; the scan shift-in patterns are altered to toggle-fill patterns when they are applied to the combinational logic, improving defect coverage.

1. Introduction

California scan architecture (CSA) is a minor modification of traditional scan architecture (TSA). Its benefits are (1) reduced power consumption during test pattern scan shift-in and (2) improved defect coverage. These benefits are made possible by applying a modified version, rather than an exact copy, of the scan shift-in pattern to the combinational logic.

This technique is feasible because most of the bits in test patterns generated by automatic test pattern generation (ATPG) tools are don't-care bits. Table 1 reports the percentage of don't-care bits for the largest I99T circuits [Corno 00] from the ITC'99 benchmark suite [Basto 00] and the Stanford ELF18 circuit [Brand 04]. The test sets are generated using a commercial ATPG tool with a maximal compaction option. The results reveal that a large percentage of bits in single stuck-at fault (SSF) and transition fault (TF) test patterns are don't-care bits. Furthermore, it has been reported that 95% to 99% of the bits in patterns for large industrial circuits are don't-care bits [Hiraide 03][Butler 04]. Examples of don't-care bit assignments are 0-fill, 1-fill, random-fill, and repeat-fill.

The *repeat-fill* method assigns don't-care bits using the last care bit. The *0-fill* technique assigns logic-0 to each don't-care bit while the *1-fill* method assigns logic-1. These don't-care bit assignment methods limit switching activity during the scan shift-in operation, although it may reduce defect coverage. On the other hand, the *random-fill* method assigns don't-care bits using pseudo-random bits. The *toggle-fill* technique assigns don't-care bits using logic-0 and logic-1 alternately. These assignments can improve defect coverage while they increase switching activity during the scan shift-in operation

Table 1. Percentage of don't-care bits

Circuit	SSF test set	TF test set
b17	86.7 %	88.3 %
b18	88.0 %	91.8 %
b19	92.5 %	95.8 %
b20	71.9 %	70.0 %
b21	74.6 %	71.1 %
b22	71.4 %	74.4 %
ELF18	83.6 %	87.0 %

In CSA, the test patterns that are shifted into scan chains have their don't-care bits assigned using the repeat-fill technique. These assignments limit the amount of toggling of scan cells during the scan shift-in operation. The patterns applied to the combinational logic are altered in the test application process so that they correspond to toggle-fill patterns. This modification of patterns enhances the defect coverage of test patterns. Table 2 shows an example of entering a scan shift-in pattern into a scan chain and applying an altered pattern to the combinational logic, in which the scan cell 1 is connected to a scan-out pin. A *test cube* is a deterministic test pattern generated by ATPG tools without assigning don't-care bits [Wang 06].

Table 2. Example of test pattern modification

Scan cell	8	7	6	5	4	3	2	1
Test cube	d	**1**	**1**	d	d	d	d	**0**
Scan-in pattern	1	**1**	**0**	0	0	0	0	**0**
Applied pattern	0	**1**	**1**	0	1	0	1	**0**

Table 3 illustrates the alteration scheme for applying the patterns. Note that the underlined entry in Table 2 differs from the value in the test cube since it will be complemented when applied to the combinational logic.

1-4244-1128-9/07/$25.00

Table 3. Example of correspondence between scan shift-in patterns and patterns applied to the combinational logic

Scan-in pattern	Q_8	Q_7	Q_6	Q_5	Q_4	Q_3	Q_2	Q_1
Applied pattern	$\overline{Q}_8$	Q_7	$\overline{Q}_6$	Q_5	$\overline{Q}_4$	Q_3	$\overline{Q}_2$	Q_1

Table 4 shows the states of scan cells during the scan shift-in operation of the pattern in Table 2. The right-most bit in the "Scan-in pattern" column enters the scan chain. During the scan shift-in operation, the repeat-fill pattern is modified to toggle-fill pattern. Note that the amount of toggling of each scan cell is limited because don't-care bits are assigned using the repeat-fill technique.

Table 4. States of scan cells during scan shift-in

Scan clock	Scan-in pattern	Scan cell 8	7	6	5	4	3	2	1
1	1100000 0	1	?	?	?	?	?	?	?
2	110000 0	1	0	?	?	?	?	?	?
3	11000 0	1	0	1	?	?	?	?	?
4	1100 0	1	0	1	0	?	?	?	?
5	110 0	1	0	1	0	1	?	?	?
6	11 0	1	0	1	0	1	0	?	?
7	1 1	0	0	1	0	1	0	1	?
8	1	0	1	1	0	1	0	1	0

CSA can improve the quality of scan-based test sets, such as SSF and TF test sets. Two techniques can be used to detect TFs in a scan-based circuit: one is skewed-load, also known as launch-on-shift [Eichelberger 91][Savir 93]; the other is broadside, also known as launch-on-capture [Eichelberger 91][Savir 94]. The TF testing in this paper uses the launch-on-capture technique to detect TFs. CSA can be used for any scan cell design, such as muxed-D scan design [Williams 73], two-port flip-flop design (clocked-scan design) [McCluskey 86], and level-sensitive scan design (LSSD) [Eichelberger 77].

This paper is organized as follows: Section 2 presents CSA and simulation results that show the effectiveness of CSA; Section 3 discusses scan shift-out switching activity; Section 4 reports experimental results that support the effectiveness of CSA; Section 5 concludes this paper; Section 6 summarizes related research.

2. California Scan Architecture

As presented in Sec. 1, most of the bits in deterministic test patterns are don't-care bits. To reduce power consumption during scan shift-in operation, don't-care bits can be assigned using the repeat-fill technique. This assignment strategy, however, may reduce the fortuitous detection of untargeted faults or defect coverage. To enhance defect coverage, don't-care bits can be assigned using the random-fill technique. But this approach increases overall power consumption during the scan shift-in operation. Therefore, in scan based testing, reducing power consumption during test application and improving defect coverage are conflicting goals. CSA provides a trade-off between test quality and power consumption by modifying test patterns during the scan shift-in operation. CSA is feasible because most of the bits in deterministic test patterns generated by ATPG tools are don't-care bits.

In CSA, scan shift-in patterns have their don't-care bits assigned using the repeat-fill method, limiting the amount of toggling of scan cells during scan shift-in operation. In the test application process, the scan shift-in patterns are modified so that the patterns applied to the combinational logic become toggle-fill patterns. Figure 1 illustrates two CSA implementations. Scan enable and clock signals are

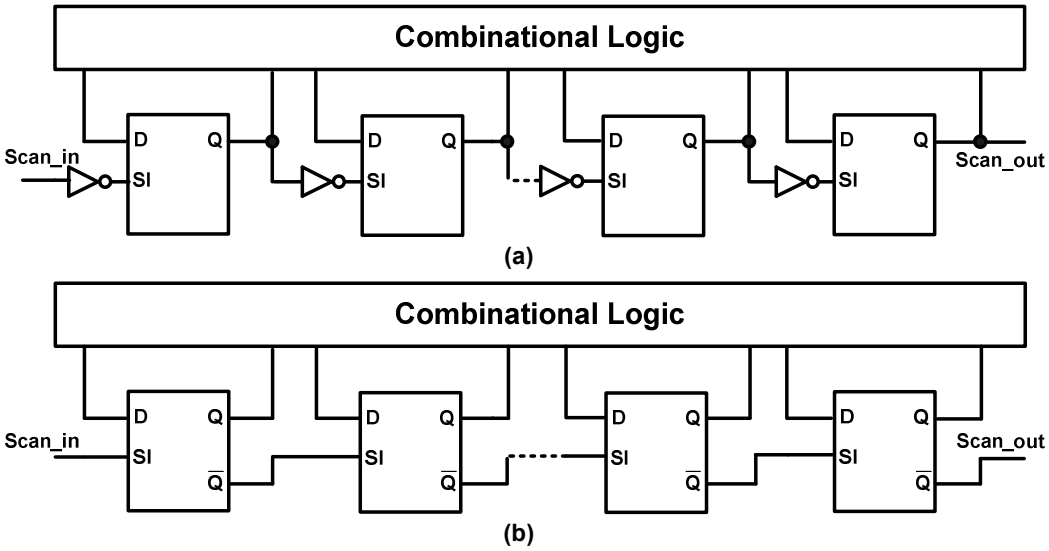

Figure 1. California scan architectures:
(a) An implementation using inverters; (b) An implementation using $\overline{Q}$ signals of scan flip-flops

not included in the figures.

Figure 1(a) shows a CSA implementation that inserts an inverter at the scan-in (SI) input of each scan flip-flop. Figure 1(b) illustrates another implementation that connects the SI input of each scan flip-flop to the $\overline{Q}$ output, rather than Q output, of the previous scan flip-flop. CSA does not change the combinational logic part of the system. In the CSA implementations, the test patterns applied to the combinational logic are different from the scan shift-in patterns. For example, when the scan shift-in pattern is "000000," the pattern applied to the combinational logic in Fig. 1(a) becomes "101010" assuming the right-most bit is the first bit to enter the scan chain. Similarly, the pattern applied to the combinational logic in Fig. 1(b) becomes "010101." One assumption of CSA is that toggle-fill patterns are more effective in detecting defects than repeat-fill patterns. Experimental results that support this assumption will be presented in Sec. 4 using the Stanford ELF18 test chips [Brand 04].

In this paper, the six largest I99T benchmark circuits [Corno 00] are used to show the implementation and effectiveness of CSA. The synthesis of the benchmark circuits and scan insertion are conducted using the Synopsys Design Compiler [Synopsys]; only one scan chain is inserted into each circuit to simplify the simulation process. After the circuits are synthesized, a PERL script inserts an inverter at the SI input of each scan flip-flop. Table 5 reports the synthesis results of the benchmark circuits: the number of scan flip-flops, primary inputs, and primary outputs.

Table 5. Benchmark circuits used in this research

Circuit	Scan flip-flops	Primary inputs	Primary outputs
b17	1,315	41	97
b18	3,016	40	24
b19	6,642	49	31
b20	430	36	23
b21	490	36	23
b22	613	36	23

Figure 2 illustrates the overall simulation flow. Test sets are generated using the Synopsys TetraMAX [Synopsys] with a maximal compaction option and without don't-care bit assignments; TSA implementations of the benchmark circuits are used to generate test sets. To apply the test sets generated for TSA netlists to corresponding CSA netlists, some care bits are complemented so that the same care bits are applied to the combinational logic (note the mapping in Table 3). Power consumption and defect detection are compared to show the effectiveness of CSA. Table 6 shows the definitions for scan architectures and don't-care bit assignments.

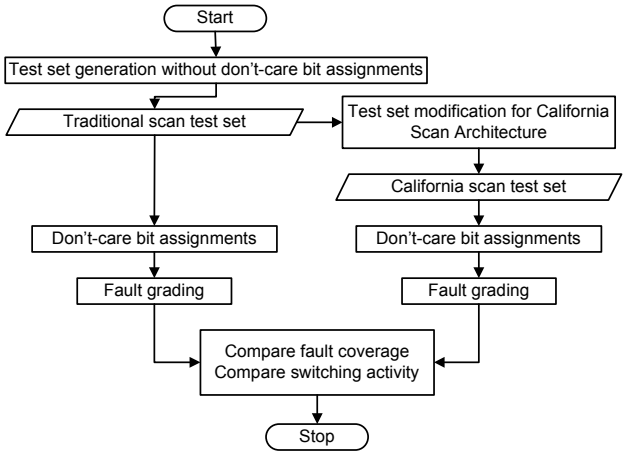

Figure 2. Simulation flow

Table 6. Scan architecture and don't-care bit assignments

Definition	Scan architecture	Don't-care bit assignments
TSA_rpt	Traditional scan	Repeat-fill
TSA_rnd	Traditional scan	Random-fill
CSA_rpt	California scan	Repeat-fill

To estimate power consumption during the scan shift operation, the number of weighted transitions (WTs) [Sankaralingam 00] is used. The method counts the switching activity of scan flip-flops during the scan shift-in and shift-out operations; a larger number represents more power consumption. It is shown that there is a close correlation between WTs and state transitions in the combinational logic [Sankaralingam 00]. Assuming the scan chain illustrated in Fig. 3, the numbers of WTs of the scan flip-flops during scan shift-in and shift-out operations are calculated using Expressions 1 and 2, respectively. In Expressions 1 and 2, x_i is a scan shift-in or scan shift-out logic value corresponding to scan flip-flop $\mathbf{x_i}$, and $\oplus$ is an exclusive-OR operation. The expressions can be applied to both TSA and CSA. To estimate capture power consumption, the number of scan flip-flops that have different logic values between applied patterns and captured responses is used.

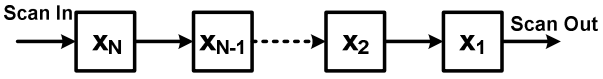

Figure 3. Example scan chain

$$WT_{\text{scan-in}} = \sum_{i=1}^{N-1} (N-i)(x_i \oplus x_{i+1}) \qquad (1)$$

$$WT_{\text{scan-out}} = \sum_{i=1}^{N-1} (i)(x_i \oplus x_{i+1}) \qquad (2)$$

The defect coverage of a test set is estimated using the average number of detected SSFs and TFs per pattern and *N*-detect coverage [Ma 95]. The use of the average number

of detected faults per pattern as a metric is based on the report that multiple detection of faults improves the defect detection of test sets [Ma 95][Grimaila 99] [Amyeen 04]. Each test pattern is fault simulated without dropping detected faults to find the average number per pattern. In this research, only the faults existing in the combinational logic are considered. All the coverage values are calculated as test coverage; i.e., untestable faults are not considered during the fault simulation.

In the case of the SSF test sets, the numbers of observed gate input combinations are also compared. An *observed gate input combination* of an internal gate is a logic combination applied to the gate inputs with the gate output being sensitized to at least one observation point, such as a primary output or a scan flip-flop [Cho 05]. The number of observed gate input combinations shows a better correlation with defect detection than the SSF coverage or the bridge coverage estimate [Cho 05][Guo 06].

Figure 4 illustrates the simulation flow using an example. The test cube used in the example for a TSA circuit is "**00**dddddd**11**," in which "d" is a don't-care bit. The test cube for the corresponding CSA circuit is "**10**dddddd**01**"; two care bits underlined are complemented. In TSA, don't-care bits are assigned using the repeat-fill and random-fill methods; in CSA, don't-care bits are assigned using only the repeat-fill technique. Figure 4 also presents scan shift-in patterns and test patterns applied to the combinational logic. In TSA, all the scan flip-flops are assigned the same logic value as in scan shift-in patterns; in CSA, alternate scan flip-flops are assigned complemented logic values of the corresponding bits in scan shift-in patterns. During test

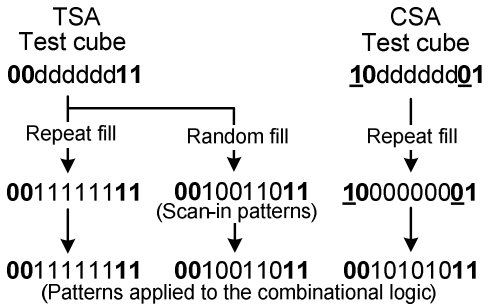

Figure 4. An example simulation flow

application, the care bits applied to the combinational logic are the same for the three configurations. Thus, the differences in the fault coverage or the number of detected faults are only the effect of don't-care bit assignments.

The switching activity for care bits can be different. For example, in Fig. 4, the scan shift-in patterns for **TSA_rpt** and **CSA_rpt** are "**00**111111**11**" and "**10**000000**01**," respectively. Assuming that the right-most bit enters a scan chain first, the numbers of WTs for **TSA_rpt** and **CSA_rpt** are 2 and 10, respectively.

Table 7 provides the simulation results for the SSF test sets. The first four columns in Table 7 present the circuit name, the configuration, the number of test patterns, and the SSF test coverage, respectively. The "Detected SSFs" column reports the average number of detected SSFs per pattern. The last column of Table 7 provides the average switching activity per pattern during test process. The average number is calculated from scan shift-in, capture, and scan shift-out switching activity. In this paper, the "Ratio" column shows the number normalized to that of **TSA_rnd**.

Table 7. Simulation results: SSF test sets

Circuit	Configuration	Test length	SSF test coverage	Detected SSFs		Observed GIC		Switching activity	
				Average	Ratio	Total	Ratio	Average	Ratio
b17	**TSA_rpt**	438	99.6	3,268	0.834	67,687	0.882	98,884	0.121
	CSA_rpt	438	99.6	3,527	**0.900**	69,643	**0.918**	195,284	**0.238**
	TSA_rnd	438	99.6	3,918	1.000	76,039	1.000	820,487	1.000
b18	**TSA_rpt**	718	99.7	7,771	0.806	186,033	0.894	493,848	0.114
	CSA_rpt	718	99.7	8,965	**0.930**	190,276	**0.914**	930,717	**0.214**
	TSA_rnd	718	99.7	9,637	1.000	208,197	1.000	4,349,002	1.000
b19	**TSA_rpt**	821	99.7	20,428	0.823	361,223	0.849	2,067,661	0.097
	CSA_rpt	821	99.7	24,014	**0.967**	368,376	**0.866**	3,298,596	**0.155**
	TSA_rnd	821	99.7	24,834	1.000	425,298	1.000	21,229,982	1.000
b20	**TSA_rpt**	692	100	1,752	0.915	36,310	0.956	22,330	0.260
	CSA_rpt	692	100	1,808	**0.944**	36,793	**0.969**	36,455	**0.424**
	TSA_rnd	692	100	1,914	1.000	38,003	1.000	86,049	1.000
b21	**TSA_rpt**	473	99.9	2,324	0.984	29,985	0.952	23,391	0.216
	CSA_rpt	473	99.9	2,234	**0.946**	30,395	**0.965**	49,156	**0.454**
	TSA_rnd	473	99.9	2,362	1.000	31,504	1.000	108,237	1.000
b22	**TSA_rpt**	711	100	2,633	0.918	55,079	0.953	45,955	0.264
	CSA_rpt	711	100	2,712	**0.945**	56,128	**0.971**	76,158	**0.437**
	TSA_rnd	711	100	2,869	1.000	57,792	1.000	174,414	1.000

The "Ratio" column gives the number normalized to that of **TSA_rnd**.

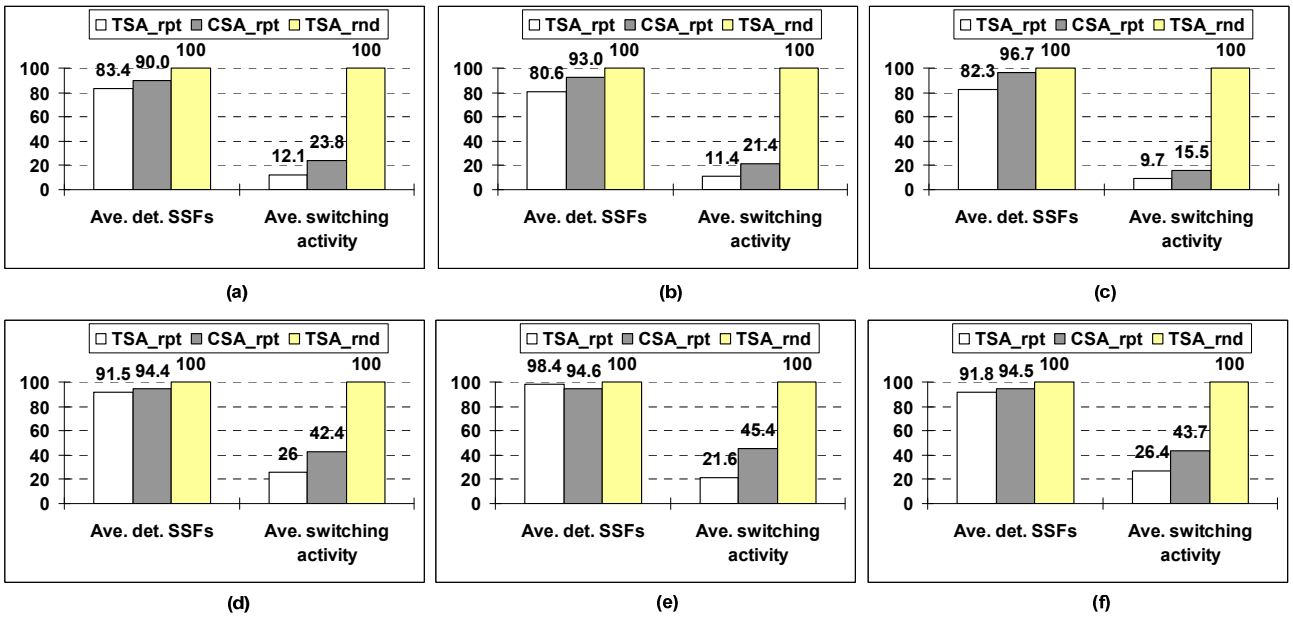

Figure 5. Comparison of average number of detected SSFs per pattern and average switching activity per pattern; the number of TSA_rnd is used as a reference: (a) b17; (b) b18; (c) b19; (d) b20; (e) b21; (f) b22

In the case of the largest benchmark circuit (b19), the average number of detected SSFs of **CSA_rpt** is 96.7 % that of **TSA_rnd** while the average switching activity is only 15.5 % that of **TSA_rnd**.

The "Observed GIC" column presents the number of observed gate input combinations obtained by gate exhaustive simulation [Cho 05]. For each benchmark circuit, **CSA_rpt** observes more gate input combinations than **TSA_rpt**, although the number is smaller than that of **TSA_rnd**.

Figure 5 compares the average number of detected SSFs per pattern and the average switching activity per pattern for each benchmark circuit. In the graphs, each number is normalized to that of corresponding **TSA_rnd**. Except the b21 circuit, the number of **CSA_rpt** is between **TSA_rpt** and **TSA_rnd**. The results show that CSA can be a good trade-off between power consumption and defect detection.

Table 8 provides the simulation results for the TF test sets.

Table 8. Simulation results: TF test sets

Circuit	Configuration	Test length	TF test coverage	Detected TFs		Switching activity	
				Average	Ratio	Average	Ratio
b17	**TSA_rpt**	991	91.0	822	0.951	96,545	0.116
	CSA_rpt	991	91.0	888	**1.028**	162,873	**0.196**
	TSA_rnd	991	91.0	864	1.000	832,254	1.000
b18	**TSA_rpt**	1,378	91.7	2,193	0.890	494,510	0.112
	CSA_rpt	1,378	91.7	2,406	**0.977**	866,492	**0.197**
	TSA_rnd	1,378	91.7	2,463	1.000	4,396,664	1.000
b19	**TSA_rpt**	2,714	95.2	4,114	0.855	1,560,428	0.073
	CSA_rpt	2,714	95.2	4,543	**0.944**	3,267,048	**0.154**
	TSA_rnd	2,714	95.2	4,814	1.000	21,234,101	1.000
b20	**TSA_rpt**	1,104	97.5	642	0.982	25,691	0.301
	CSA_rpt	1,104	97.5	649	**0.992**	40,473	**0.474**
	TSA_rnd	1,104	97.5	654	1.000	85,464	1.000
b21	**TSA_rpt**	677	96.5	745	0.949	30,310	0.282
	CSA_rpt	677	96.5	775	**0.988**	56,357	**0.525**
	TSA_rnd	677	96.5	785	1.000	107,379	1.000
b22	**TSA_rpt**	1,333	97.7	943	0.992	48,196	0.275
	CSA_rpt	1,333	97.7	924	**0.973**	74,356	**0.425**
	TSA_rnd	1,333	97.7	950	1.000	175,145	1.000
The "Ratio" column gives the number normalized to that of **TSA_rnd**.							

The "TF test coverage" and "Detected TFs" columns report the TF test coverage and the average number of detected TFs per pattern, respectively. Except the b22 circuit, the average number of the detected TFs per pattern of **CSA_rpt** is larger than that of **TSA_rpt**. Figure 6 demonstrates that the increases in the number of detected TFs of **CSA_rpt** from **TSA_rpt** are higher than the increases in power consumption. For example, in the case of the b19 circuit, the average number of the detected TFs of **TSA_rpt** is 85.5% that of **TSA_rnd**, which increases to 94.4% by applying CSA; on the other hand, the power consumption increases from 7.3% to 15.4% that of **TSA_rnd**.

The switching activity of **CSA_rpt** is larger than that of **TSA_rpt** because many care bits are clustered, and the same logic value is assigned to the clusters of care bits. In the CSA circuits, alternate care bits are complemented in scan shift-in patterns, which may increase switching activity during the scan shift-in operation.

Table 9 reports the *N*-detect test coverage [Ma 95] for the SSF test sets of I99T benchmark circuits. The overall trends show that the *N*-detect coverage of **CSA_rpt** is between that of **TSA_rpt** and **TSA_rnd**.

Don't-care bits can be assigned using care bits during test set generation; this strategy will reduce the number of test patterns by detecting untargeted faults fortuitously. Table 10 reports the results of b19 SSF test sets for the three configurations. The number in parenthesis represents the ratio to the number in **TSA_rnd**. Figure 7 demonstrates that CSA increases the average number of detected SSFs from 83% to 97.9% that of **TSA_rnd** while it increases power consumption only from 10.6% to 17.8%.

Table 9. *N*-detect coverage: SSF test sets

Circuit	Conf.	2-det	5-det	10-det	15-det
b17	**TSA_rpt**	73.5	57.5	45.2	35.3
	CSA_rpt	73.4	57.7	45.9	36.5
	TSA_rnd	73.6	59.3	47.5	37.5
b18	**TSA_rpt**	75.8	59.5	45.7	36.8
	CSA_rpt	75.9	59.8	46.2	37.3
	TSA_rnd	77.4	63.6	49.7	40.8
b19	**TSA_rpt**	72.4	57.2	45.8	39.4
	CSA_rpt	72.4	58.2	46.4	39.8
	TSA_rnd	73.9	61.0	50.2	43.7
b20	**TSA_rpt**	77.8	53.4	34.7	29.3
	CSA_rpt	78.4	53.0	35.4	29.7
	TSA_rnd	78.3	53.8	35.5	29.5
b21	**TSA_rpt**	78.2	57.1	42.6	38.3
	CSA_rpt	78.1	58.2	44.2	39.0
	TSA_rnd	78.5	58.2	44.3	39.5
b22	**TSA_rpt**	78.4	54.5	36.0	30.7
	CSA_rpt	79.0	55.2	36.1	30.4
	TSA_rnd	79.0	54.8	36.3	30.4

The simulation results show that CSA is more effective in the b19 circuit than other benchmark circuits. The b19 circuit is the largest among the benchmark circuits, and the test sets for the circuit contain the largest percentage of don't-care bits. This shows that the effectiveness of CSA may increase when it is applied to circuits that contain more don't-care bits in their patterns than the small benchmark circuits used in this research.

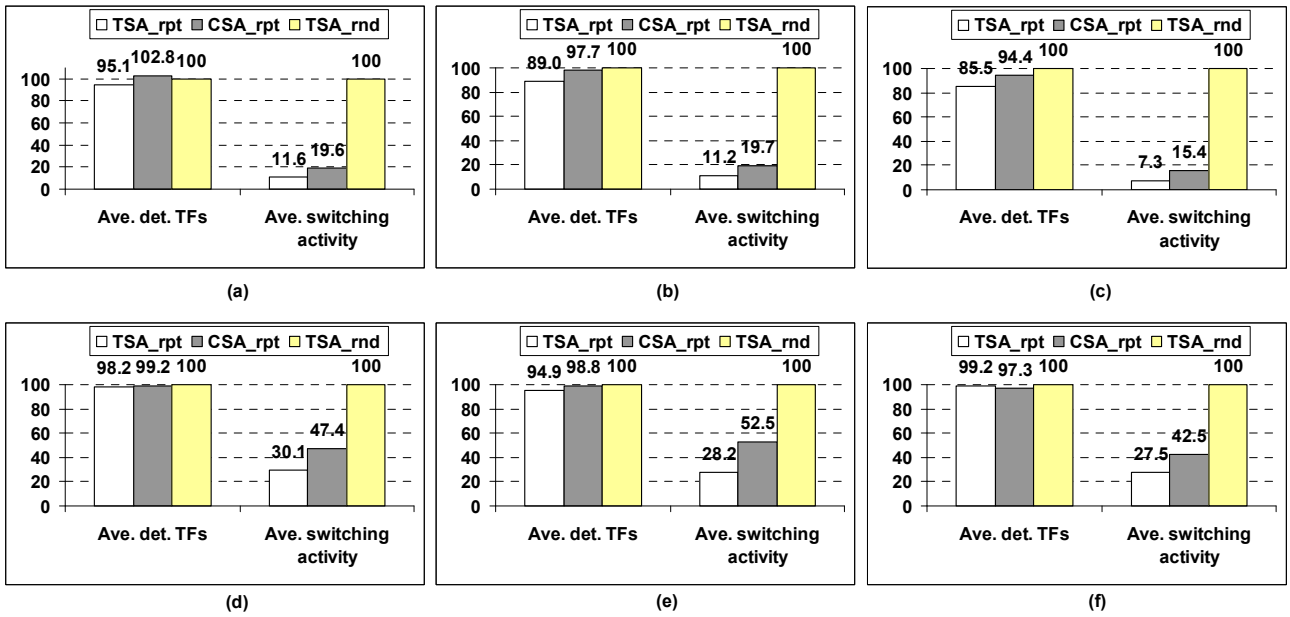

Figure 6. Comparison of average number of detected TFs per pattern and switching activity per pattern; the number of TSA_rnd is used as a reference: (a) b17; (b) b18; (c) b19; (d) b20; (e) b21; (f) b22

Table 10. SSF test generation results with don't-care bits being assigned during test generation (b19)

	TSA_rpt	CSA_rpt	TSA_rnd
SSF test coverage	99.7 %	99.7 %	99.7 %
Test length	575	565	591
Ave. det. SSFs	20,146 (0.830)	23,765 (**0.979**)	24,283 (1.000)
Ave. Scan-in activity	2,246,454 (0.106)	3,772,448 (**0.178**)	21,188,266 (1.000)

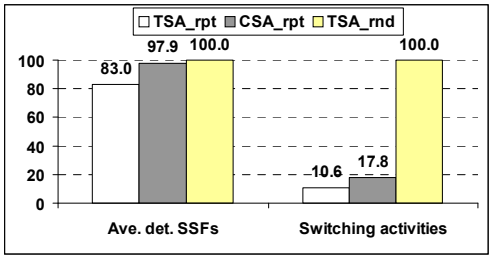

Figure 7. Average number of detected SSFs per pattern and average switching activity per pattern (b19); the number is normalized to that of TSA_rnd

3. Scan Shift-Out Switching Activity

CSA controls the power consumption during the scan shift-in operation, whereas the power consumption during the scan shift-out operation depends on the captured patterns, which are determined by the applied patterns and the circuit structure. This section further investigates power consumption during the scan shift-out operation.

Table 11 reports scan shift-in and shift-out switching activity of the SSF test sets for the I99T benchmark circuits. In the case of the b19 circuit, the percentages of switching activity of **TSA_rpt** and **CSA_rpt** during the scan shift-in operation are 7.3% and 12.3% that of **TSA_rnd**, respectively; the percentages of switching activity of **TSA_rpt** and **CSA_rpt** during the scan shift-out operation are 12.3% and 18.8%, respectively. Figure 8 illustrates the scan shift-in and shift-out switching activity normalized to the total switching activity of **TSA_rnd**. The simulation results reveal that switching activity during the scan shift-out operation follows a trend similar to switching activity during the scan shift-in operation.

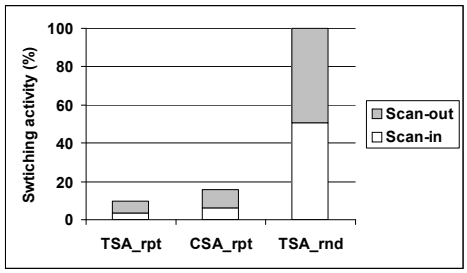

Figure 8. Comparison of switching activity: b19 SSF test sets

Table 11. Scan-in and scan-out switching activity: SSF test sets

Cir.	Config.	Scan-in		Scan-out	
		Average	Ratio	Average	Ratio
b17	**TSA_rpt**	38,587	0.093	60,180	0.149
	CSA_rpt	70,329	**0.169**	124,819	**0.309**
	TSA_rnd	416,901	1.000	403,464	1.000
b18	**TSA_rpt**	209,924	0.096	283,649	0.131
	CSA_rpt	386,293	**0.177**	544,163	**0.252**
	TSA_rnd	2,184,849	1.000	2,163,898	1.000
b19	**TSA_rpt**	782,128	0.073	1,284,906	0.123
	CSA_rpt	1,328,231	**0.123**	1,969,828	**0.188**
	TSA_rnd	10,764,761	1.000	10,464,679	1.000
b20	**TSA_rpt**	9,106	0.209	13,164	0.310
	CSA_rpt	14,458	**0.332**	21,942	**0.517**
	TSA_rnd	43,544	1.000	42,450	1.000
b21	**TSA_rpt**	9,403	0.168	13,902	0.266
	CSA_rpt	17,523	**0.313**	31,570	**0.604**
	TSA_rnd	55,944	1.000	52,231	1.000
b22	**TSA_rpt**	18,309	0.208	27,555	0.319
	CSA_rpt	30,327	**0.345**	45,750	**0.530**
	TSA_rnd	87,953	1.000	86,379	1.000

One explanation of the trend is that not all scan flip-flops change states between applied patterns and captured patterns. Table 12 reports the percentage of scan flip-flops that have different logic values between the applied patterns and the captured patterns for the SSF test sets. The percentage is less than 17.6%. Therefore, there is a correlation between scan shift-in patterns and scan shift-out patterns in the benchmark circuits.

Table 12. Percentage of the time that scan flip-flops have different logic values between the applied patterns and captured patterns: SSF test sets

Circuit	TSA_rpt	CSA_rpt	TSA_rnd
b17	8.9	10.3	9.3
b18	9.1	8.7	8.5
b19	9.4	8.1	8.2
b20	14.0	12.8	12.8
b21	17.6	12.7	12.7
b22	14.8	13.2	13.4

To show the dependency of scan shift-out switching activity on the synthesis library, the b19 circuit is synthesized using a different technology library. Table 13 reports the switching activity of an SSF test set. The results show that the switching activity of **CSA_rpt** during scan shift-out operation is 27.8% that of **TSA_rnd**. The switching activity normalized to the total switching activity of **TSA_rnd** is presented in Fig. 9. The percentage of scan shift-out switching activity is a little higher than the result in Table 11; however, the overall trend in switching activity during the scan shift operation is similar as presented in Figs. 8 and 9.

Table 14 provides the average number of detected SSFs per pattern for the circuits synthesized using the different technology library. In this case, the average number of **CSA_rpt** is larger than that of **TSA_rnd**.

Table 13. Scan shift switching activity: b19 SSF test sets

Average switching activity	TSA_rpt	CSA_rpt	TSA_rnd
Scan-in	750,392	1,643,218	8,657,669
Scan-out	1,123,262	2,393,407	8,597,605
Total	1,873,654	4,036,625	17,255,274

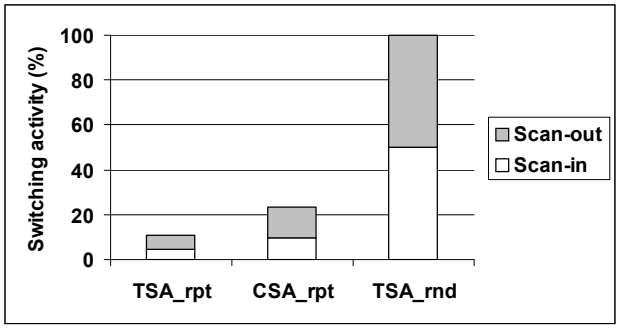

Figure 9. Comparison of switching activity: b19 SSF test sets

Table 14. Average number of detected SSFs per pattern

Configuration	Detected SSFs per pattern	
	Average	Ratio
TSA_rpt	27,409	0.895
CSA_rpt	30,719	1,003
TSA_rnd	30,633	1.000

The switching activity of SSF test sets for the ELF18 core is also compared. The percentages of scan flip-flops that change states during the capture period are 15.0%, 22.1%, and 22.3% for **TSA_rpt**, **CSA_rpt**, and **TSA_rnd**, respectively. Table 15 shows that the average switching activity of **CSA_rpt** during the scan shift-out operation is 83.8% that of **TSA_rnd** for the ELF18 SSF test sets. In the case of the ELF18 SSF test sets, the average numbers of switching activity of **TSA_rpt** and **CSA_rpt** during the scan shift operation are 15% and 48% that of **TSA_rnd**, respectively. Figure 10 illustrates the relative switching activity normalized to the total switching activity of **TSA_rnd**.

Table 15. Scan shift switching activity: an ELF18 SSF test set

Average switching activity	TSA_rpt	CSA_rpt	TSA_rnd
Scan-in	9,428	18,135	99,737
Scan-out	20,781	67,602	80,696
Total	30,209	85,737	180,433

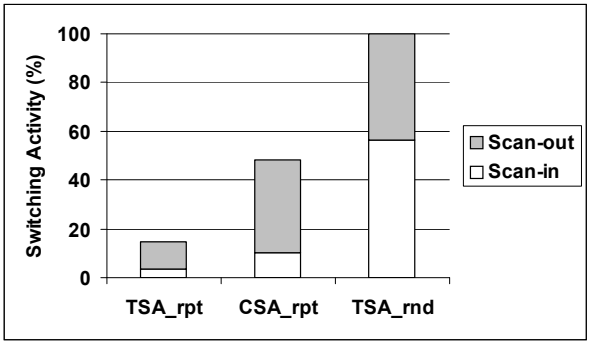

Figure 10. Comparison of switching activity: ELF18 SSF test sets

The simulation results reveal that CSA reduces switching activity during the scan shift-in operation. Even though switching activity during the scan shift-out operation depends on the circuit structure, the simulation results show that CSA also reduces scan shift-out switching activity. In scan-based testing, the scan shift-in and shift-out operations are conducted concurrently. Therefore, total switching activity during the scan shift operation can be reduced by reducing scan shift-in switching activity, even though the reduction in scan shift-out switching activity is not significant.

4. Experimental Results

The Stanford ELF18 test chips [Brand 04] are used to compare the correlation between defect detection and don't-care bit assignments. The ELF18 test chips were fabricated using the Philips 0.18 micron Corelib technology, and the nominal supply voltage is 1.8V. More than 70,000 test chips were manufactured; each test chip contains 6 R.E.A.L.^{TM} digital signal processor cores. One ELF18 core contains 13 scan chains; the total number of scan flip-flops is 2,290; the total number of gates for one core is 53,732.

A SSF test set and a TF test set are generated with a maximal compaction option and without don't-care bit assignments using the Synopsys TetraMAX [Synopsys]. Then, the test sets have their don't-care bits assigned using three techniques: repeat-fill, toggle-fill, and random-fill. Table 16 describes the test sets.

Table 16. Test sets: ELF18

Test set	Test set type	Don't-care bit assignments
SSF_rpt	Single stuck-at fault	Repeat-fill
SSF_tgl	Single stuck-at fault	Toggle-fill
SSF_rnd	Single stuck-at fault	Random-fill
TF_rpt	Transition fault	Repeat-fill
TF_tgl	Transition fault	Toggle-fill
TF_rnd	Transition fault	Random-fill

Tables 17 and 18 report the simulation results of the SSF test sets and the TF test sets, respectively. The "Ratio" column shows the average number of detected faults per pattern normalized to that of **SSF_rnd** or **TF_rnd**. The trend of the ELF18 simulation results is similar to that of I99T benchmark circuit results. One exception is the fact that the average number of SSFs detected by **SSF_tgl** is larger than that of SSFs detected by **SSF_rnd**.

Table 17. SSF test set simulation results: ELF18

Test set	Test length	SSF coverage [%]	Detected SSFs	
			Average	Ratio
SSF_rpt	457	99.5	9,940	0.821
SSF_tgl	457	99.5	12,684	**1.048**
SSF_rnd	457	99.5	12,102	1.000

Table 18. TF test set simulation results: ELF18

Test set	Test length	TF coverage [%]	Detected TFs	
			Average	Ratio
TF_rpt	1,472	97.2	1,741	0.694
TF_rnd	1,472	97.2	2,503	**0.997**
TF_tgl	1,472	97.2	2,511	1.000

All test sets are applied at nominal voltage (1.8V). In the case of the TF test sets, the launch/capture clock speed is a speed with 10% margin from the maximum operation speed determined from the shmoo plots of defect-free cores. In this experiment, 1,604 cores are tested.

Figure 11 illustrates the fallout of defective cores. The experiments reveal that the random-fill and toggle-fill assignments are more effective in detecting defects than the repeat-fill assignments. Note that the difference in the defect detection among the test sets is only the effect of don't-care bit assignments. The experimental results using the ELF18 test chips demonstrate that CSA is more effective in detecting defects than **TSA_rpt** because CSA applies toggle-fill patterns to the combinational logic.

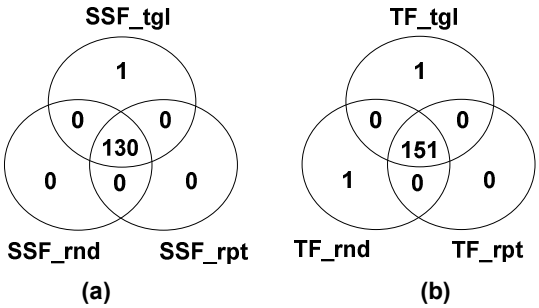

Figure 11. Detection of defective cores: (a) SSF test sets; (b) TF test sets

5. Conclusions

This paper presents California scan architecture (CSA) that achieves high quality and low power testing by modifying test patterns during the scan shift-in operation. CSA is feasible because most of the bits in test patterns generated by ATPG tools are don't-care bits.

CSA can be constructed by slightly modifying traditional scan architecture. Furthermore, CSA can be used without changing current design tools. The effectiveness of CSA with respect to power consumption and defect detection may increase when it is applied to circuits that contain more don't-care bits in their patterns than the benchmark circuits used in this paper.

6. Appendix: Review of Related Research

In modern VLSI design, full-scan design is usually used for better controllability and observability of internal states. Widely used scan designs are the muxed-D scan design [Williams 73] and the level-sensitive scan design (LSSD) [Eichelberger 77].

During the scan shift operation, the states of scan flip-flops change, causing many state transitions; these state transitions may cause good chips to fail the applied test set because of abnormal power consumption or excessive power/ground noise compared to normal operation [Nicolici 02][Sinanoglu 03][Yoshida 03]. To overcome the abnormal switching activity during the scan shift operation, an "enhanced" scan architecture [DasGupta 81] can be used to prevent the switching activity from propagating to the combinational logic while the architecture increases hardware cost. Cost-efficient implementations of gating logic to mask the switching activity during the scan shift operation are presented [Gerstendorfer 99][Bhunia 05]. The insertion of gating logic, however, introduces an additional delay in the normal operation. To reduce the maximum switching activity, scan chains are divided into multiple segments, and one segment is activated at a time [Saxena 01].

Scan chains are modified by inserting inverter and XOR gates into specific locations in scan chain paths to reduce switching activity during the scan shift operation; the scan-in data are transformed according to the modified structure [Sinanoglu 03]. This method changes scan chain paths for a specific test set that contains only care bits while CSA utilizes don't-care bit assignments to improve defect detection and to reduce power consumption during testing.

ATPG tools generate test patterns with many don't-care bits; the percentage of don't-care bits for compacted test sets is between 95% and 99% [Hiraide 03][Butler 04]. Don't-care bit assignments can be utilized to compact test sets or to reduce power consumption during testing. Examples of don't-care bit assignments are 0-fill, 1-fill, random-fill, and repeat-fill. The repeat-fill method is an efficient method with respect to test set size and power consumption during test application [Butler 04]. On the

other hand, don't-care bit assignments also affect the defect detection of test sets [McCluskey 04].

Some previous methods are test set dependent, i.e., scan chains are modified to reduce power consumption for a specific test set. In production testing, many test sets are applied; sometimes different test sets are generated after the design is completed. Moreover, scan flip-flops cannot be reordered freely because of the layout constraints [Makar 98][Bonhomme 03]. Therefore, test set dependent methods are not practical solutions in production testing. CSA does not depend on a specific test set or scan flip-flop reordering. Furthermore, CSA can be used with other scan architectures and scan design methods such as scan cell stitching and reordering.

Acknowledgment

This research was supported partially by NSF under contract number CCR-0098128 and partially by LSI Logic.

The authors would like to express their thanks to Bill Price, Stefan Eichenberger, and Fred Bowen of NXP for providing the ELF18 test chips. The authors also thank Erik Volkerink of Verigy for providing tester time.

References

[Amyeen 04] Amyeen, M. E., S. Venkataraman, A. Ojha, and S. Lee, "Evaluation of the Quality of N-Detect Scan ATPG Patterns on a Processor," *Proc. Intl. Test Conf.*, pp. 669-678, 2004.

[Basto 00] Basto, L., "First Results of ITC'99 Benchmark Circuits," *IEEE Design & Test of Computers*, vol. 17, no. 3, pp. 54-59, Jul.-Sep. 2000.

[Bhunia 05] Bhunia, S., H. Mahmoodi, D. Ghosh, S. Mukhopadhyay, and K. Roy, "Low-Power Scan Design Using First-Level Supply Gating," *IEEE Trans. on VLSI Systems*, vol. 13, no. 3, pp. 384-395, Mar. 2005.

[Bonhomme 03] Bonhomme, Y., P. Girard, L. Guiller, C. Landrault, and S. Pravossoudovitch, "Efficient Scan Chain Design for Power Minimization During Scan Testing Under Routing Constraint," *Proc. Intl. Test Conf.*, pp. 488-493, 2003.

[Brand 04] Brand, K. A., S. Mitra, E. H. Volkerink, and E. J. McCluskey, "Speed Clustering of Integrated Circuits," *Proc. Intl. Test Conf* ., pp. 1128-1137, 2004.

[Butler 04] Butler, K. M., et al., "Minimizing Power Consumption in Scan Testing: Pattern Generation and DFT Techniques," *Proc. Intl. Test Conf.*, pp. 355-364, 2004.

[Cho 05] Cho, K. Y., S. Mitra, and E. J. McCluskey, "Gate Exhaustive Testing," *Proc. Intl. Test Conf.*, Paper 31.3, 2005.

[Corno 00] Corno, F., M. S. Reorda, and G. Squillero, "RT-level ITC'99 Benchmarks and First ATPG Results," *IEEE Design & Test of Computers*, vol. 17, no. 3, pp. 44-53, Jul.-Sep. 2000.

[DasGupta 81] DasGupta, S., R. G. Walther, T. W. Williams, and E. B. Eichelberger, "An Enhancement to LSSD and Some Applications of LSSD in Reliability, Availability, and Serviceability," *Proc. Intl. Symp. on Fault-Tolerant Computing*, pp. 32-34, 1981.

[Eichelberger 77] Eichelberger, E. B. and T. W. Williams, "A Logic Design Structure for LSI Testability," *Proc. Design Automation Conf.*, pp. 462-468, 1977.

[Eichelberger 91] Eichelberger, E. B., E. Lindbloom, J. A. Waicukauski, and T. W. Williams, "Delay-Fault Simulation," *Structured Logic Testing*, E. J. McCluskey (Ed.), Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1991.

[Gerstendorfer 99] Gerstendorfer, S. and H.-J. Wunderlich, "Minimized Power Consumption for Scan-Based BIST," *Proc. Intl. Test Conf.*, pp. 77-84, 1999.

[Grimaila 99] Grimaila, M. R. et al., "REDO - Random Excitation and Deterministic Observation – First Commercial Experiment," *Proc. VLSI Test Symp.*, pp. 268-274, 1999.

[Guo 06] Guo, R., et al., "Evaluation of Test Metrics: Stuck-at, Bridge Coverage Estimate and Gate Exhaustive," *Proc. VLSI Test Symp.*, pp. 66-71, 2006.

[Hiraide 03] Hiraide, T., et al., "BIST-Aided Scan Test—A New Method for Test Cost Reduction," *Proc. VLSI Test Symp.*, pp. 359-364, 2003.

[Ma 95] Ma, S. C., P. Franco, and E. J. McCluskey, "An Experimental Chip to Evaluate Test Techniques Experiment Results," *Proc. Intl. Test Conf.*, pp. 663-672, 1995.

[Makar 98] Makar, S., "A Layout-Based Approach for Ordering Scan Chain Flip-Flops," *Proc. Intl. Test Conf.*, pp. 341-347, 1998.

[McCluskey 86] McCluskey, E. J., *Logic Design Principles: With Emphasis on Testable Semicustom Circuits*, Prentice Hall, Englewood Cliffs, NJ, 1986.

[McCluskey 04] McCluskey, E. J., et al., "ELF-Murphy Data on Defects and Test Sets," *Proc. VLSI Test Symp.*, pp. 16-22, 2004.

[Nicolici 02] Nicolici, N. and B. M. Al-Hashimi, "Multiple Scan Chains for Power Minimization during Test Application in Sequential Circuits," *IEEE Trans. on Computers*, vol. 51, no. 6, pp. 721-734, Jun. 2002.

[Sankaralingam 00] Sankaralingam, R., R. R. Oruganti, and N. A. Touba, "Static Compaction Techniques to Control Scan Vector Power Dissipation," *Proc. VLSI Test Symp.*, pp. 35-40, 2000.

[Savir 93] Savir, J. and S. Patil, "Scan-Based Transition Test," *IEEE Trans. on CAD*, vol. 12, no. 8, pp. 1232-1241, Aug. 1993.

[Savir 94] Savir, J. and S. Patil, "On Broad-Side Delay Test," *Proc. VLSI Test Symp.*, pp. 284-290, 1994.

[Saxena 01] Saxena, J., K. M. Butler, and L. Whetsel, "An Analysis of Power Reduction Techniques in Scan Testing," *Proc. Intl. Test Conf.*, pp. 670-677, 2001.

[Sinanoglu 03] Sinanoglu, O. and A. Orailoglu, "Modeling Scan Chain Modifications for Scan-in Test Power Minimization," *Proc. Intl. Test Conf.*, pp. 602-611, 2003.

[Synopsys] Synopsys, Inc., http://www.synopsys.com.

[Wang 06] Wang, L.-T., C.-W. Wu, and X. Wen (Eds.), *VLSI Test Principles and Architectures*, Morgan Kaufmann Publishers, San Francisco, CA, 2006.

[Williams 73] Williams, M. J. Y. and J. B. Angell, "Enhancing Testability of Large-Scale Integrated Circuits via Test Points and Additional Logic," *IEEE Trans. on Computers*, vol. C-22, no. 1, pp. 46-60, Jan. 1973.

[Yoshida 03] Yoshida, T. and M. Watai, "A New Approach for Low Power Scan Testing," *Proc. Intl. Test Conf.*, pp. 480-487, 2003.