Dynamic Scan Testing: Investigating A New Paradigm

Clay Gloster, Jr.*

Technical Report TR93-06

April 2, 1993
Technical Report TR93-06

* Department of Electrical and Computer Engineering
P. O. Box 7911
North Carolina State University
Raleigh, NC 27695

Dynamic Scan Testing:
Investigating a New Design Paradigm

Clay Gloster

ABSTRACT

The cost of testing can be divided into two major components: test generation and test application. For a given part, test generation is a one-time cost, while the cost of test application is expended repeatedly. Therefore, in order to minimize the total cost of testing, the optimum strategy is one that minimizes the cost of test generation as well as the test sequence length. This report presents a strategy which attempts to reduce both the cost of generating a test as well as the time required to apply this test.

The key to dynamic scan testing is the fact that we supplement an existing sequential circuit, called the object machine, with a scannable test machine to improve controllability and observability of internal machine states. Another key is a novel dynamic algorithm for test generation which takes advantage of the knowledge of the test machine hardware to reduce test generation costs and the overall length of the test sequence.

This report introduces a new cellular scan test machine which is found to perform well within the dynamic testing paradigm. It is a test machine which connects the flip-flops into one or more circular chains with each chain containing an exclusive-or gate at the beginning of the chain. One input of each exclusive-or, is directly controllable from the circuit inputs.

The report also presents a dynamic scan test generation algorithm, DYNASTEE, which reduces the *average* test sequence length per fault when compared to existing static test generation algorithms for scan architectures. It allows the mode control pin to cycle freely between object and test machine operation as required. The scan register is shifted for minimal time to reach a state required for testing. Test sequence length is further optimized via a scan chain ordering algorithm.

This report demonstrates the flexibility of dynamic scan testing by presenting it's effectiveness for both partial and multiple scan chains as well as introducing a technique for random test generation. The report concludes with results of experiments which validate the theory that dynamic scan testing can significantly reduce the cost of testing and is a viable alternative to traditional static scan testing approaches.

1 Introduction

This report contains excerpts from the MCNC Technical Report entitled, "Dynamic Scan Testing: Investigating a New Design Paradigm". It also contains information concerning the new ISCAS89 ADDENDUM benchmark circuits. These circuits have been gathered from both university and commercial researchers in the area of design and test. They provide an interesting challenge to research efforts of the future and we hope that this report will assist the users of these benchmarks.

2 The ISCAS89 ADDENDUM benchmark set

The new extension to the ISCAS89 benchmark set consists of 14 circuits all containing no combinationally redundant faults. Table 1, shown below, provides detailed information about the circuits. The original along with the suggested name is give for each circuit. The suggested names were developed in the same fashion as the ISCAS89 benchmark set using an "s" followed by a number. The "s" identifies the circuit as sequential and the number corresponds to the number of lines in the circuit. The number of inputs, outputs, and flip-flops for each benchmark is also shown in Table 1. The number of inputs for the circuits range from 1 to 83 and the number of outputs from 1 to 73. No benchmark contains less than 15 flip-flops and the largest benchmark contains 239. The number of single stuck-at faults in the collapsed fault set for each circuit is also shown in the column labeled "Faults".

The final two columns in the table identify the properties of the combinational logic of the benchmarks or the sequential circuit with all flip-flops removed. The column labeled "Random FC(%)" illustrates the fault coverage of the combinational logic block of the circuits after 65,536 random test patterns have been applied. While the circuits s344, s967, s1269, and s6669 are testable in less than 65,536 random trials, the circuit s499 is the most resistant to testing with random patterns. The final column in the table provides the number of test vectors/patterns required in a test set that would detect 100% of the faults in the combinational logic. This is not necessarily the minimum sized set.

Table 1: Static Benchmark Characteristics in Full Scan Mode

Original Name	Suggested Name	Inputs	Outputs	Flip-Flops	Faults	FC (%) after 65536 Trials	ATPG Vecs (FC = 100%)
s344	s344	9	11	15	342	100.00	27
pair22	s499	1	22	22	583	48.20	128
hard32	s635	2	1	32	666	73.72	74
s838.1	s938	34	1	32	931	66.81	177
cc1	s967	16	23	29	1066	100.00	109
false	s991	65	17	19	910	97.03	42
s1196	s1196	14	14	18	1242	99.92	148
alu	s1269	18	10	37	1343	100.00	59
barcode	s1512	29	21	57	1357	95.36	87
fpunit	s3271	26	14	116	3270	99.94	133
prolog	s3330	40	73	132	2870	89.09	316
mult	s3384	43	26	183	3380	96.21	106
design2	s4863	49	16	104	4764	99.03	76
bound	s6669	83	55	239	6684	100.00	59

The ISCAS89 ADDENDUM benchmark set provides several key challenges. While the following list is far from exhaustive, examples of potential uses for the ISCAS89 addendum benchmark set include:

1. Investigation of multiple scan chain configurations while reporting minimum test application time for achieving 100% fault coverage.

2. Investigation of a combination of multiple and partial scan where a subset of all of the circuit flip-flops are included in several chains while reporting minimum test application time for achieving 100% fault coverage.

3. Generation of partitions and orderings of flip-flops and the effects on test application time.

4. Minimum-length test set generation using new sequential test generation algorithms while maintaining 100% fault coverage.

5. Identification and/or removal of sequentially redundant faults.

6. Improvement of (weighted) random testing strategies for both combinational and sequential circuits while maintaining 100% fault coverage.

3 Static versus Dynamic Scan

One of the key objectives of our experiments[1, 2, 3] was to reduce the cost of test application. In support of the objective, we made comparisons between a traditional static test generation algorithm and a new dynamic test generation algorithm using a subset of the new benchmark set. In our experiments, a single traditional or cellular scan chain was added to each benchmark and test generation was performed. We set out to identify whether the static approach, or the proposed dynamic approach would produce a set of test patterns that detect 100% of the single stuck-at faults with minimum test application costs.

While the reader may not be familiar with cellular scan, it is very similar to traditional scan. Traditional scan connects the flip-flops of a sequential circuit into a shift register in the test mode. The input and output of the shift register are the *Scanin* and *Scanout* pins respectively. Unlike traditional scan, cellular scan connects the flip-flops of the sequential circuit into a circular shift register. An exclusive-or gate combines the output of the last flip-flop of the shift register with the *Scanin* pin. While cellular scan does not include a *Scanout* pin, fault coverages of 100% can still be obtained.

The key premise of dynamic scan test generation is that the algorithm adjusts itself during generation of test sequences. Each test sequence consists of a sub-sequence for fault excitation followed by a sub-sequence for fault propagation. Based on the current state of the machine, the test sequence generation selects the next target fault which has the ***shortest length excitation sequence***. In fact, the excitation state required for a particular fault could be compatible with the current state of the machine. In that case, the sequence will have minimal length. The algorithm also computes the ***shortest length propagation sequence*** to minimize test sequence length. On average, the length of dynamically generated test sequences can be much smaller than those of the traditional static test sequence generation techniques.

One benefit of the results of our research efforts is the fact that we can identify circuits which will benefit the most from the dynamic approach prior to the process of test generation. An important indicator of the speedup we can expect to achieve with a dynamic strategy for a given circuit is comparing the *average length* of the excitation *and* propagation sequences to the total number of flip-flops in the chain. See [1, 3] for details of calculation of average test sequence length and other dynamic characteristics. Tables 2 and 3 show the anticipated speedup over static scan for each of the circuit benchmarks. First, a set of three valued excitation and propagation vectors were generated using combinational test generation for each circuit benchmark. The magnitude of the set of excitation vectors, V_e and the magnitude of the set of propagation vectors, V_p, is recorded in both tables. The density of literals of the excitation and propagation vectors, ρ_e, and ρ_p provide a good first order indication of potential speedup when using the dynamic approach. Circuits whose excitation and propagation vectors have low densitites will have the largest potential speedup. Notably, many large circuits exhibit this property.

Given the natural order of the scan chain extracted from the circuit netlist, the average number of clock cycles required for fault excitation, $\overline{n}_e$, and fault propagation, $\overline{n}_p$, are computed. The sum of $\overline{n}_e$ and $\overline{n}_p$ is the average number of clock cycles to test an arbitrary fault, $\overline{n}$, or the average test sequence length. All three of these parameters are presented in the tables for each circuit benchmark. Potential speedup in test application time is the ratio of the number of flip-flops, n, to the average test sequence length, $\overline{n}$. It seems that the circuit $s3384$ has the largest potential speedup of the complete benchmark set.

At first glance, it appears that Table 2 contains the same information as the Table 3, but there is a subtle difference. In Table 2, dynamic circuit characteristics were computed for all faults in the original circuit benchmark. In Table 3, fault simulation was performed on the benchmark circuit in the normal mode of operation for 65,536 clock cycles and the dynamic characteristics were computed for only the remaining subset of faults.

Table 2: Dynamic Benchmark Characteristics

Original	Suggested			Estimated Densities		Avg. Clock Cycles (Projected)				
Name	Name	V_e	V_p	ρ_e	ρ_p	$\overline{n}_e$	$\overline{n}_p$	$\overline{n}$	n	Speedup
s344	s344	377	11	0.24	0.04	5	1	6	15	2.50
pair22	s499	633	22	0.62	0.00	14	1	15	22	1.47
hard32	s635	738	32	0.17	1.00	6	31	37	32	0.86
s838.1	s938	663	32	0.27	0.90	10	17	27	32	1.19
cc1	s967	1180	23	0.15	0.00	4	2	6	29	4.83
false	s991	432	19	0.12	0.02	3	1	4	19	4.75
s1196	s1196	329	16	0.03	0.06	2	2	4	18	4.50
alu	s1269	1392	28	0.16	0.01	17	1	18	37	2.06
barcode	s1512	1497	42	0.08	0.07	11	10	21	57	2.71
fpunit	s3271	4124	14	0.04	0.00	21	8	29	116	4.00
prolog	s3330	2642	56	0.06	0.02	33	4	37	132	3.57
mult	s3384	4123	63	0.02	0.04	17	2	19	183	9.63
design2	s4863	5290	32	0.07	0.01	37	2	39	104	2.67
bound	s6669	6960	110	0.03	0.03	65	2	67	239	3.57

4 Test Generation Results

In the interest of timely release of the new benchmark set, we present results for 9 of the 14 circuit benchmarks. A future publication will present similar results for the remaining circuits.

In our experiments, we included a single cellular scan chain to each circuit benchmark. A single *Select* pin was added to distinguigh between test mode (shifting data into the scan chain) and normal circuit operation. The *Scanin* pin was shared with an arbitrary primary input pin. Combinational redundancies which appeared in any multiplexors of the scan chain were removed prior to test generation to avoid fault coverages of less than 100%.

Table 4 illustrates the effects of scan chain order on test application costs when using the dynamic test generation approach. All of the circuits achieve 100% fault coverage with test application cost being measured by counting the the number of clock cycles required. We considered two full cellular scan single-chain orderings of flip-flops. The initial ordering is based on the order of flip-flops in the original circuit netlist. It is an arbitrary or random ordering of the flip-flops of the scan chain which we call the natural order. The optimized order is based on the ordering algorithm discussed in [2, 3]. The ordering minimizes the

Table 3: Dynamic Benchmark Characterisitics after 65,536 clock cycles in normal mode

Original Name	Suggested Name	V_e	V_p	Estimated Densities ρ_e	ρ_p	Avg. Clock Cycles (Projected) $\overline{n}_e$	$\overline{n}_p$	$\overline{n}$	FFs	Speedup
s344	s344	48	12	0.12	0.04	3	1	4	15	3.75
pair22	s499	211	22	0.71	0.00	16	1	17	22	1.29
hard32	s635	730	32	0.17	0.97	6	16	22	32	1.45
s838.1	s938	395	32	0.34	0.87	12	15	27	32	1.19
cc1	s967	104	23	0.11	0.00	4	2	6	29	4.83
false	s991	63	19	0.09	0.01	3	1	4	19	4.49
s1196	s1196	61	16	0.05	0.06	2	2	4	18	4.50
alu	s1269	115	29	0.04	0.01	3	1	4	37	9.25
barcode	s1512	764	43	0.10	0.07	12	10	22	57	2.59
fpunit	s3271	359	14	0.02	0.00	7	8	15	116	7.73
prolog	s3330	1073	56	0.08	0.02	43	4	47	132	2.81
mult	s3384	701	64	0.01	0.04	6	2	8	183	22.88
design2	s4863	542	32	0.04	0.01	22	2	24	104	4.39
bound	s6669	717	110	0.01	0.03	9	2	11	239	21.73

projected average test sequence length $\overline{n}$. In most cases, the value of $\overline{n}$ is a pessimistic estimate of the actual average test sequence length $\overline{n}_{act}$ which we measure directly upon completion of dynamic test generation by dividing the total number of clock cycles required by the total number of faults. Notably, improving the projected average test sequence length $\overline{n}$ also improves the total test sequence length measured in clock cycles by about 23% for all circuits. The clock cycles designated with an asterisk (*) indicate cases where no exclusive-or gate is required in the chain, i.e. the shift register is not circular and the output of the last flip-flop is left floating in the test mode.

Our results show that the number of clock cycles required for 100% fault coverage of 9 of the 14 benchmark circuits can be reduced by 51% when compared to the static test compaction techniques. The results of this experiment is shown in Table 5. A traditionally compacted scan test of the 9 circuits would require 27,534 clock cycles for 100% fault coverage. This is due to the fact that 891 vectors generated by an ATPG program must be shifted serially via the scan chain.

Using our new dynamic scan test generation algorithm (DYNASTEE), we avoid shifting data completely into the shift register. While the number of test sequences (which is comparable to the number of ATPG vectors generated) for our approach was larger (1,181), the total number of clock cycles required to apply the dynamic test was only 13,525.

With the dynamic algorithm, in conjunction with a scan chain ordering algorithm, the average test sequence length can be reduced to a value which is less than n. For the 9 benchmarks selected , the average number of flip-flops is 29. The average test sequence length for traditional scan is $n + 1$, where n is the number of flip-flops. Our experimental results show that we reduced the average test sequence length from (27534/891=30.90) for the traditional static scan test generation approach to (13,525/1,181=11.45) for the dynamic scan test generation approach.

5 Test Patterns Subdirectory

A subdirectory (ISCAS89/ADDENDUM93/PATTERNS) contains a set of test patterns and scan chain configurations which have been used to achieve 100% fault coverage. This directory contains two subdirectories: TSCAN and CSCAN. The TSCAN directory contains the test patterns for the circuit benchmarks with a

Table 4: Test generation with cellular scan: initial and final order of flip-flops

Original Name	*Suggested Name*	**Natural Order** $\overline{n}$	$\overline{n}_{act}$	*Clock cycles*	**Optimized Order** $\overline{n}$	$\overline{n}_{act}$	*Clock cycles*
s344	s344	6	5	192	5	4	154
pair22	s499	12	9	1269*	12	9	1269*
hard32	s635	44	43	1830	44	43	1830
s838.1	s938	30	26	4605	28	27	4843
cc1	s967	11	7	899*	6	6	970*
false	s991	5	3	149	4	3	139*
s1196	s1196	3	3	499	3	3	419
alu	s1269	18	8	1041	10	5	641*
barcode	s1512	28	30	7633	10	14	3569*
Total		157	134	18117	122	114	13834

* denotes that the exclusive-or feedback is not required

Table 5: Comparing test generation: cellular scan with static traditional scan

Original Name	*Suggested Name*	**All algorithms achieve 100% fault coverage** — **Total Cycles** *DYNASTEE Cell. Scan*	*Comb. ATPG Trad. Scan*	**Total Sequences** *DYNASTEE Cell. Scan*	*Comb. ATPG Trad. Scan*
s344	s344	154	511	42	31
pair22	s499	1269	3104	148	134
hard32	s635	1830	2144	43	64
s838.1	s938	4605	5345	178	161
cc1	s967	899	3629	138	120
false	s991	139	939	55	46
s1196	s1196	419	3058	187	160
alu	s1269	641	2773	132	72
barcode	s1512	3569	6031	258	103
Total		13525	27534	1181	891

single chain traditional scan test machine and the CSCAN directory contains patterns for a single cellular scan test machine. All test pattern files were obtained using the dynamic scan test generation approach. In some cases, sequential fault simulation will provide additional compaction, however the extent of this compaction is negligible. The challenge is to find test pattern sets that achieve 100% fault coverage with minimum test application costs, where test application costs is measured by the number of clock cycles required to apply the test.

A sample test pattern file for the circuit *s*344 follows. Each file contains a header and the actual patterns separated by a dotted line. Lines which contain a comment begin with a "#" sign. The header lists the name of the circuit and the primary inputs and outputs. The topmost primary input is assigned the leftmost bit of each test pattern and a single test pattern is applied for a single clock cycle. The order of the remaining primary inputs in the header of the file determine the remaining test pattern bit assignments during each clock cycle. The number of primary inputs and outputs are also listed. Outputs patterns, also included in the file, are all set to zero as they are ignored. Note that two additional inputs are shown for each benchmark: RESET and Select. The RESET input clears all flip-flops of the circuit in a single clock cycle. The Select input is the control input of all multiplexors included in the scan chain.

Portion of File: s344.tp

```
.N s344

.I na="RESET"
.I na="Select"
.I na="START"
.I na="B0"
.I na="B1"
.I na="B2"
.I na="B3"
.I na="A0"
.I na="A1"
.I na="A2"
.I na="A3"

.O na="P4"
.O na="P5"
.O na="P6"
.O na="P7"
.O na="P0"
.O na="P1"
.O na="P2"
.O na="P3"
.O na="CNTVCON2"
.O na="CNTVCO2"
.O na="READY"


.i 11
.o 11

#-----------------------------------------------------
# Fault name: BMVSON->BMVG5VG1VAD1NF /1 BE
# Excitation Sequence length: 1
# Propagation Sequence length: 1
```

```
# Test Sequence length: 2
10101001101 00000000000
01011101110 00000000000
# Fault name: CT2->INIT /0 BE
# Excitation Sequence length: 1
# Propagation Sequence length: 1
# Test Sequence length: 2
00000100110 00000000000
01101011001 00000000000
# Fault name: AX1->AD1N /1 BE
# Excitation Sequence length: 1
# Propagation Sequence length: 1
# Test Sequence length: 2
00010000111 00000000000
01011011000 00000000000
# Fault name: CNTVG3VZ1 /1 BE
# Excitation Sequence length: 1
# Propagation Sequence length: 1
# Test Sequence length: 2
00001101111 00000000000
01111001010 00000000000
# Fault name: CT0->INIT /0 BE
# Excitation Sequence length: 1
# Propagation Sequence length: 2
# Test Sequence length: 3
00001110010 00000000000
01110000011 00000000000
00010001111 00000000000
```

The scan chain configuration file is also included for each circuit benchmark. It can be used to describe various single/multiple partial/full scan chain configurations and allows benchmark users to identify the scan chain order. The first line of the file tells how many scan chains were used. Since all of the current experiments contained a single scan chain, all current chain files contain a 1 in the first line. The following lines show how many flip-flops are in each chain and the order of the flip-flops for each chain. The reference order is the order of the flip-flops in the original circuit benchmark description and numbering of flip-flops begins with 0. A sample chain file is shown below.

File: s344.chn

```
1
15 2 1 10 0 4 5 12 3 13 8 7 14 6 11 9
```

The natural chain order for the above example corresponding to the order of the flip-flops contained in the original circuit netlist is 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14.

The chain file can also be used to describe multiple and partial scan chains. An example of a scan chain configuration with several chains and a few non-scannable flip-flops follows. In the example, there are three scan chains, the first of length 3, the second of length 6, and the third of length 2. These chains include the flip-flops on lines 2-4 of the file. The last line tells which flip-flops are not to be included in the scan chain. Here there are 4 flip-flops, number 9, 10, 13, and 14 which are non-scannable.

File: mult.partial.chn

```
3
3 0 2 4
6 1 3 5 6 7 12
2 8 11
4 9 10 13 14
```

References

[1] C. Gloster and F. Brglez. Synthesis of Weighted Random Test Sequences with Application to Testing of Sequential Circuits. In *European Conf. on Design Automation (EDAC'93)*, pages 231–237, Feb. 1993.

[2] C. Gloster and F. Brglez. Cellular Scan Test Generation for Sequential Circuits. In *European Design Automation Conference (EURO-DAC '92)*, pages 530–536, September 1992.

[3] C. Gloster. *Dynamic Scan Testing: A New Paradigm.* PhD thesis, North Carolina State University, Raleigh, NC, March 1993. Also published as: MCNC Technical Report TR93-06, Dynamic Scan Testing: Investigating a New Design Paradigm, Microelectronics Center of North Carolina, Research Triangle Park, N.C., April 1993.