First Results of ITC'99 Benchmark Circuits

Luis Basto
Analog Devices

The ISCAS circuits have long been used as design-for-testability benchmarks; however, recent progress in technology requires newer DFT standards. This look at the ITC'99 benchmarks reveals complexities that will serve as a starting point for other researchers.

■ **BENCHMARKS ARE IMPORTANT** vehicles that let industry and academia develop new tools, compare and contrast different methodologies, and research new algorithms and techniques. In design-for-testability (DFT) work, the standard sets have been the International Symposium on Circuits and Systems (ISCAS) '85 and '89 circuits.[1,2] These circuits have been in longtime use over many years and much theoretical work has been done using them as a basis. However, evolution of both design styles and tools calls for new, larger, and more complex circuits that more closely represent current technology.

Table 1 lists the set of circuits in the ITC'99 benchmark portfolio (http://www.cerc.utexas.edu/itc99-benchmarks/bench.html). It includes several designs from industry, the Torch design from Stanford University, the Carnegie Mellon University digital signal processor (CMUDSP) design, and a group of circuits from Polytechnic University of Torino.

This article examines the I991, I992, and I99C1 circuits in some detail because they are already in structural Verilog format and gives a brief overview of 1994 and 1995 circuits. One requirement for the ITC'99 benchmarks was that they be written in either of two hardware description languages—Verilog or VHSIC hardware description language (VHDL)—since most electronic design automation (EDA) tools, as well as some freeware simulators, support them. All the circuits are available in register transfer level (RTL) Verilog except the Torino benchmarks, which are in VHDL. Although it would be desirable to provide both Verilog and VHDL equivalents for all benchmarks, as Table 1 shows, these circuits are available in either, but not both. To do so would entail the additional task of verifying that the Verilog and VHDL netlists are logically equivalent, a worthwhile undertaking that will make the benchmarks more useful to a wider audience.

Baseline characteristics

The baseline characteristics of these circuits appear in Table 2. I994 (the Torch design) is synthesized into logic gates and provided as a reference to gauge the complexity of the other circuits. All data gathered for this study are from industrial tools, such as those from Mentor Graphics Corp. and Synopsys Inc.

Numbers shown in this article are absolute; for example, the number of faults is the total number of faults before fault equivalence collapsing. Not all fault categories are shown—the "possibly detected" and "possibly undetected" fault categories were not listed, for instance, so the numbers do not add up to 100%. My testing considered only the input and output faults of logic gates. For example, a 3-input complex and-or-invert gate might consist of several logical Boolean primitives, but the only faults I considered are the three inputs and one output, or a

0740-7475/00/$10.00 © 2000 IEEE

Table 1. ITC'99 benchmark circuits.

Design	Description	Format
I99C1	Combinational circuit extracted from industrial design	Structural Verilog
I991	Two-phase latch-based design of ~6,000 gates	Structural Verilog
I992	Four-phase ASIC design of ~20,000 gates	Structural Verilog
I993	ASIC design with embedded memories, multiple clock domains (Not ready for release)	Structural Verilog
I994 (Torch)	Superscalar microprocessor modeled on MIPS R2000/R3000 instruction set	Behavioral RTL Verilog
I995 (CMUDSP)	CMU DSP modeled after Motorola DSP56002 (Needs work on library and removal of scan chain)	Behavioral and structural Verilog
I99T	22 sequential circuits of fair to high complexity with some industrial design subsets	Synthesizable VHDL

Table 2. Base properties of subset of ITC'99 circuits.

Design	Inputs	Outputs	Bidirectional	Sequential Cells	Total Faults
I99C1	1,288	8	0	0	6,356
I991	133	256	14	769	34,919
I992	207	264	176	1,784	103,008
I993	72	32	30	—	—
I994 (Torch)	126	108	96	5,133	112,147
I995 (CMUDSP)	104	101	61	1,990	110,362

total of 8 uncollapsed stuck-at faults.

All of these designs are hierarchical. Theoretically, flattened and hierarchical designs—which are more of an issue in design flows, methodology, and CAD tools—should not affect testability. However, flattened designs have higher resource requirements and need more memory, disk space, and CPU time to run.

Analysis of I99C1

I99C1 is the only combinational circuit in the set and is relatively straightforward. Automatic test-pattern generation (ATPG) tools quickly generated 92.1% fault coverage; greater coverage is not possible because of tied and redundant faults. The uncontrollable category also includes 24 untested faults consisting of these six and their equivalents:

```
/F4T2/AD1/VA/UU5/AN1/OUT
/F4T2/AD2/VA/UU5/AN1/OUT
/F4T2/AD1/VA/UU6/AN1/OUT
/F4T2/AD2/VA/UU6/AN1/OUT
/FINADD/VA/UU5/AN1/OUT
/FINADD/VA/UU6/AN1/OUT
```

They all have large input cones of logic, more than 15 levels, tracing back to 16 primary inputs with massive, reconvergent fanout. Increasing time and abort limits fail to detect any of these faults, which may be good study candidates for tuning ATPG algorithms.

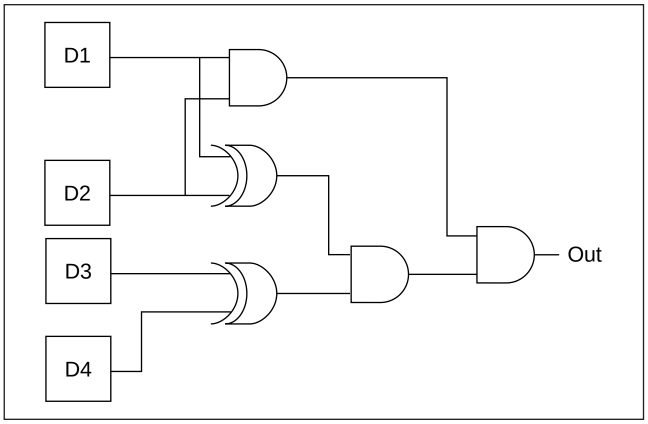


Figure 1. An output signal always tied to logic zero in this I99C1 example.

I99C1 resembles a 32-bit adder built with cascading 2-bit half-adders. Due to the way it is implemented, a large number of signals are permanently at logic zero. Figure 1 shows one example. The OUT node of the AND gate is always stuck at logic 0 regardless of the values on inputs D1, D2, D3, and D4. This one tied fault contributed to a number of other blocked faults in both backward and forward logic cones. All

> With full scan inserted, the ATPG tool fails to trace the scan chain. Simply replacing sequential cells with their scan equivalents does not make a design testable or create a functional scan chain.

the tied faults are stuck at zero, and I99C1 has many such faults.

As an experiment, I synthesized the design to see the effect of eliminating these faults. I expected that the synthesis tool would, through optimization, eliminate the tied signals and perhaps give better coverage. Table 3 shows the unexpected results. The synthesis tool eliminated untested faults and reduced redundant faults, but it also created many tied signals, which resulted in slightly decreased fault coverage. Perhaps different synthesis constraints would reduce the number of tied signals and other faults, but that lies outside the scope of this article.

Analysis of I991 and I992

The industry designs are much more interesting because of their greater complexity and existence of design rule violations. Table 4 shows their base characteristics.

I obtained the fault coverages shown here from running a full-scan ATPG tool on the original circuits, without any test logic such as scan insertion. The extremely low coverages were due to a full-scan ATPG tool's inability to generate tests for complex nonscan circuits. Partial-scan ATPG tools, which I did not use, would provide a bit more coverage.

The method to analyze these designs follows:

- obtain base fault coverage of the raw circuit,
- insert full scan,
- run combinational ATPG tools to get fault coverage, and
- insert additional test logic as necessary to increase coverage.

When trying to insert scan into these designs, I encountered one problem these benchmarks are likely to present. Although the two circuits are inherently Verilog gates, they are not supplied in structural Verilog format. I991 was also missing a basic gate library. While names such as NAND2 and OR3 might be intuitively meaningful to humans, they must be defined for EDA tools. As a result of this work, I have submitted a corrected design with a logic gate library to the benchmark site.

Test methods

The most common scan methodologies in use today are multiplexed scan and level-sensitive scan design (LSSD). When full scan is implemented in a design, the fault coverage is relatively independent of the scan methodology. Multiplexed scan, which is what I used for this analysis, is easier to implement and is faster for the ATPG tools to generate tests for, as well.

Scan insertion is implemented by replacing sequential elements, namely flip-flops and latches, with their scannable equivalents and connecting these into a serial shift register with primary scan inputs and outputs. With full scan inserted, the ATPG tool still fails to trace the scan chain. Simply replacing sequential cells with their scan equivalents does not make a design testable or, in this case, create a functional scan chain.

The blockage results from two issues. First, the only gated latch in the design, XTCSTALL/XTCEIMS/I15485, has a small cone of logic driving the Clock Enable signal. I experimented with bypassing this signal and making this one latch nonscan. Making the latch nonscan essentially resulted in a partial-scan circuit and caused a slight drop in coverage. Therefore, to fix the scan chain, I chose to bypass the Enable. Second, two nets that are outputs from latches, therefore scan-out ports, also happen to be primary bidirectional pins. The industrial ATPG tool I used in this study is unable to handle bidirectionality during scan

Table 3. Comparison of results for I99C1.

Circuit Tested	Total Faults	Detected Faults	Tied Faults	Redundant Faults	Untested	Fault Coverage (%)
Original circuit	6,356	5,852	384	96	24	92.1
Resynthesized circuit	8,784	8,028	720	36	0	91.4

Table 4. Characteristics of a subset of I99x circuits.

Benchmark	Combined loops	Total faults	Tied/blocked faults	Redundant faults	Unused	ATPG untestable	Fault coverage (%)
I991	1	34,919	1,868	62	1,486	30,754	1.8
I992	17	103,008	1,710	5	2,052	98,743	0.5
I994 (Torch)	67	112,147	2,716	2	17,028	92,207	0.2
I995 (CMUDSP)	99	110,362	338	500	1,154	108,370	0.0

Table 5. Test results of I991.

Benchmark	Total Faults	Detected Faults	Tied Faults	Redundant Faults	Untested	ATPG Untestable	Fault Coverage (%)	Test Coverage (%)	Number of Test Patterns
I991S1	37,236	33,270	1,867	1,763	1	131	89.4	99.5	847
I991S2	37,472	33,931	1,676	1,577	1	50	90.7	99.7	855

chain tracing. I made a small change and relabeled these two nets, MUTDAS and XT2YB, as outputs. Another possibility is adding separate scanout ports for these two signals, bypassing the bi-directional pins. In summary, to make this circuit testable in this instance, two changes were bypassing a gated latch and forcing two bidirectional ports as outputs.

The above techniques are comparable to what would be done in industry to make designs testable. It would be interesting to see if other tools or researchers can run ATPG without making these alterations.

Test results

Table 5 shows the test coverage results of circuit I991 with the modifications mentioned above. I991S1 is the almost-full-scan circuit. I991S2 is the same scan design with the addition of 20 test points for enhanced controllability and 20 test points for enhanced observability. The ATPG tool chose and inserted the test points automatically.

The untested column is the number of faults that ATPG has not yet attempted, so some of these may be detected with increased test generation effort. The fault coverage percentage contains the absolute number considering all fault categories. The test coverage percentage is generally the industry-accepted number: the number of detected faults divided by the total faults minus the untestable faults, including tied, blocked, and redundant faults.

For I991, the number of tied and redundant faults is almost 10% of the total, resulting in a fault coverage of roughly 90%, while the test coverage is close to 100%. A significant number of test points will have to be added to make fault coverage approach test coverage.

I992 is a latch design driven by a four-phase clock. When inserting scan, 160 of the 1,784 latches were not scannable due to nontransparency. The result is therefore a partial-scan design. Of the 160 nonscan latches, 158 were due to gated clocks, and the other two were actually flip-flops used as set and clear signals.

Table 6. Test results of I992.

	Total	Detected	Tied	Redundant		ATPG	Coverage	Coverage	Test
Benchmark	**Faults**	**Faults**	**Faults**	**Faults**	**Untested**	**Untestable**	**(%)**	**(%)**	**Patterns**
I992S1	107,928	63,280	1,354	580	1,042	26,821	64.9	66.9	1,488
I992S2	108,372	87,394	1,432	818	1,124	3,532	86.5	89.5	2,772
I992S3	108,382	87,414	1,434	818	1,129	3,596	85.7	88.7	2,552
I992S4	108,620	88,023	1,434	808	1,115	3,151	86.9	89.9	2,801

These gated clocks are predominantly in modules WIDGET/WREGS, WIDGET/LOADDUMP, SEQ, and SEQ/STACK. After tying off the nonsensitizing inputs to these gated clocks, thereby making it a full-scan design, I achieved better coverage, as expected.

Table 6 shows the results. I992S1 is the partial-scan circuit. I992S2 is the almost full-scan circuit will all latches scanned except the 2 flip-flops. I992S3 is the full-scan circuit including the two set/clear flip-flops. Finally, I992S4 is full scan with an additional 20 test points to increase controllability and 20 to increase observability, similar to what was done on I991.

Some interesting observations arise from the I992 benchmark. Apparently forcing the set and clear inputs of the two flip-flops to make them scannable resulted in a slight increase in untestable faults.

Another simplification was to ignore bus contentions due to multiple drivers and bidirectional signals. This also establishes the upper limit of the fault coverage that can be achieved for this circuit. I attempted to analyze the 333 potential bus contentions that can occur during ATPG. One of the main contributors is IAPATH20_Inst/ I1323_Inst, which is one of many drivers on a 20-bit bus. However, the inputs of this block are all tied to logic zero. Therefore, whenever this driver is enabled, it has a very high probability of causing bus contention. By disabling this driver and constraining the other drivers to be one-hot, I eliminated most of the bus contentions. By then, the fault coverage dropped to about 73% with a test coverage of 75%.

Inserting more test points resulted only in a slight increase in fault coverage. A large number of untestable faults remains due to many signals tied to power and ground. Many of these are inputs to multiplexers, so the best achievable fault coverage for this design will not be very high.

Preliminary analysis of I994 and I995

A minor effort was made to analyze these two designs. Tables 2 and 4 show that they are comparable in size to I992 but they are quite a bit more complex. I994 was synthesized with minimal optimizations to obtain a gate-level netlist. Like I991, it is a two-phase latch-based design. It is, however, far more complex as it is a full processor chip with memory caches and register files, which must be modeled as black boxes for test purposes. It contains tri-state buses with more than 700 opportunities for bus contention.

I995, which contains no fewer than 13 clocks, is a mixed latch and flip-flop design. It has the most combinational loops of all the benchmark designs. While on the surface it appears to contain a scan chain, I found no scan cells in the design, leaving the impression that a scan chain was removed, yet some of the scan signals remain. It does, however, contain tri-state buses and transistor primitives.

THE DATA PRESENTED here represent an initial attempt to understand the ITC'99 benchmark circuits and to provide a base data set for evaluating them. Future researchers can use some of the information outlined here to plan their work.

The number of tied and redundant faults on I991 is almost 10%, limiting the top achievable fault coverage to about 90%. Test logic or test points can be added, as I have done to increase the coverage, but there is a point of diminishing returns. Ignoring bus contentions, I992 has a large number of hard-to-test and ATPG-untestable faults that also caps its coverage at around 90%. When bus contentions are taken into account, the coverage dropped to about 73%. The ATPG tool cuts combinational loops on both designs.

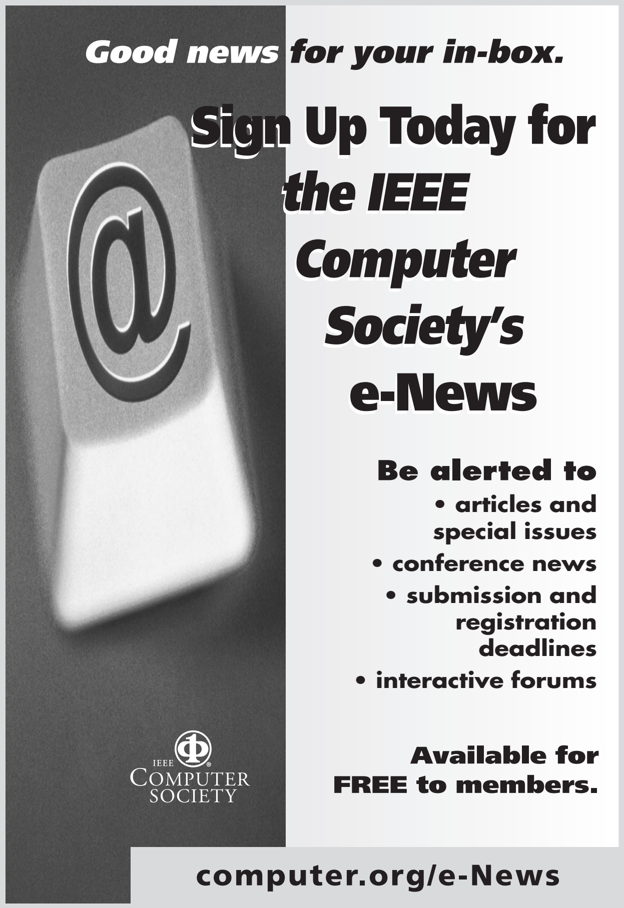
There was a software quip that said, “If it is hard to program, it should be hard to read.” A similar corollary for hardware might be, “If it is hard to design, it should be hard to test.” It does not have to be that way. Regardless, these benchmarks should provide ample fuel for further research and analysis in the years ahead. ■

■ References

1. F. Brglez and H. Fujiwara, “A Neutral Netlist of 10 Combinational Benchmark Circuits,” *Proc. IEEE Int’l Symp. Circuits and Systems*, IEEE Press, Piscataway, N.J., 1985, pp. 695-698.
2. F. Brglez, D. Bryan, and K. Kozminski, “Combinational Profiles of Sequential Benchmark Circuits,” *Proc. IEEE 1989 Int’l Symp. Circuits and Systems*, IEEE Press, Piscataway, N.J., May 1989, pp. 1924-1934.

Luis Basto is a Project Lead for DFT in the DSP Division at Analog Devices in Austin, Texas. He received his BSEE for the University of Houston and his MSEE from Southern Methodist University. His research interests include design and test methodology, automated test development and diagnosis, and System-on-Chip testing. Basto is a member of the IEEE.

■ You may address comments and questions regarding this article to Luis Basto, Analog Devices, Inc., 8911 N. Capital of Texas Hwy., Suite 1300, Austin, TX 78759-7203, Luis.Basto@analog.com.



Good news for your in-box.
Sign Up Today for the IEEE Computer Society’s e-News
Be alerted to
• articles and special issues
• conference news
• submission and registration deadlines
• interactive forums
Available for FREE to members.
IEEE Computer Society
computer.org/e-News