Test Cost Minimization for Hybrid BIST[1]

Gert Jervan
Department of Computer and Information Science, Linköping University, SE-581 83 Sweden
E-mail: gerje@ida.liu.se

Zebo Peng
Department of Computer and Information Science, Linköping University, SE-581 83 Sweden
E-mail: zebpe@ida.liu.se

Raimund Ubar
Department of Computer Engineering, Tallinn Technical University, Estonia
E-mail: raiub@pld.ttu.ee

Abstract

This paper describes a hybrid BIST solution for testing systems-on-chip which combines pseudorandom test patterns with stored deterministic test patterns. A method is proposed to find the optimal balance between pseudorandom and stored test patterns to perform core test with minimum time and memory, without losing test quality. Two accurate algorithms are proposed for finding the optimal time-moment to stop pseudorandom test generation and to apply stored patterns. To speed up the optimization procedure, a method is proposed for fast estimation of the expected cost for different possible solutions with very low computational cost. Experimental results have demonstrated the feasibility of the proposed approach for cost optimization of hybrid BIST.

1. Introduction

The rapid advances of the microelectronics technology in recent years have brought new possibilities to integrated circuits (ICs) design and manufacturing. Many systems are nowadays designed by embedding predesigned and preverified complex functional blocks, usually referred as cores, into one single die. The cores can be very different by their nature (from analog to memories, including all types of logic) and can be represented in several different ways (RTL code, netlist or layout) [10]. Such a design style allows designers to reuse previous designs and will lead therefore to a shorter time to market and a reduced cost. Such a System-on-Chip (SoC) approach is very attractive from the designers' perspective. Testing of SoC, on the other hand, shares all the problems related to testing modern deep submicron chips, and introduces also some additional challenges due to the protection of intellectual property as well as the increased complexity and higher density [6].

To test the individual cores of the system the test pattern source and sink have to be available together with an appropriate test access mechanism (TAM) [12]. We can implement such a test architecture in several different ways. A widespread approach implements both source and sink off-chip and requires therefore the use of external Automatic Test Equipment (ATE). But, as the internal speed of SoC is constantly increasing and the technology used in ATE is always one step behind, the ATE solution will soon become unacceptably expensive and inaccurate [7], leading also to an unacceptable yield loss. Therefore, in order to apply at-speed tests and to keep the test

[1] This work was partially supported by EC Copernicus project VILAB, the Swedish National Board for Industrial and Technical Development (NUTEK), the Swedish Academy of Engineering Sciences, and the Estonian Science Foundation (Grants No 3658, 4003).

costs under control, on-chip test solutions are needed. Such a solution is usually referred to as built-in self-test (BIST).

A BIST architecture demands a test pattern generator (TPG), a test response analyzer (TRA) and a BIST control unit (BCU). The classical way to implement the TPG and TRA for BIST is to use linear feedback shift registers (LFSRs). However, the LFSR-based approach often does not guarantee sufficiently high fault coverage (especially in the case of large and complex designs) and demands very long test application times. Therefore, several proposals have been made to combine pseudorandom test patterns, generated by LFSRs, with deterministic patterns [3], [4], [8], [11], to form a hybrid BIST solution.

The main concern of the hybrid BIST approaches has been to improve the fault coverage by mixing pseudorandom vectors with deterministic ones, while the issue of cost minimization has not been addressed directly. The time minimization problem is addressed in [8], where the main concern is to minimize the time during test scheduling by selecting appropriate test scenarios for individual cores.

The main objective of the current work is to find the optimal balance between pseudorandom and deterministic test patterns to perform core test with minimum cost of both time and memory, without losing test quality. We propose in this paper an algorithm to estimate, with very low computational time, the optimal time-moment to stop pseudorandom test generation and to apply deterministic patterns.

The rest of the paper is organized as follows. In section 2 we introduce the concepts of hybrid BIST; section 3 gives an overview of test cost minimization for hybrid BIST together with cost estimation method; and in section 4 we present the experimental results which demonstrate the feasibility of our approach. In section 5 we will draw some conclusions together with an introduction to future work.

2. Hybrid BIST

As test patterns generated by LFSRs are pseudorandom by nature, the generated test sequences are usually very long and not sufficient to detect all the faults. To avoid the test quality loss due to random pattern resistant faults and to speed up the testing process, we have to apply deterministic test patterns targeting the random resistant and difficult to test faults. Such a hybrid BIST approach starts with a pseudorandom test sequence of length L. On the next stage, stored test approach takes place. For the stored test approach, precomputed test patterns, stored in the system, are applied to the core under test to reach the desirable fault coverage. For off-line generation of S deterministic test patterns, arbitrary software test generators may be used, based on deterministic, random or genetic algorithms.

In a hybrid BIST technique the length of the pseudorandom test is an important parameter, which determines the behavior of the whole test process. A shorter pseudorandom test set implies a larger deterministic test set. This requires additional memory space, but at the same time, shortens the overall test process. A longer pseudorandom test, on the other hand, will lead to longer test application time with reduced memory requirements. Therefore it is crucial to determine the optimal length of pseudorandom test in order to minimize the total testing cost.

Figure 1 illustrates graphically the total cost of a hybrid BIST consisting of pseudorandom test patterns and stored test patterns, generated off-line.

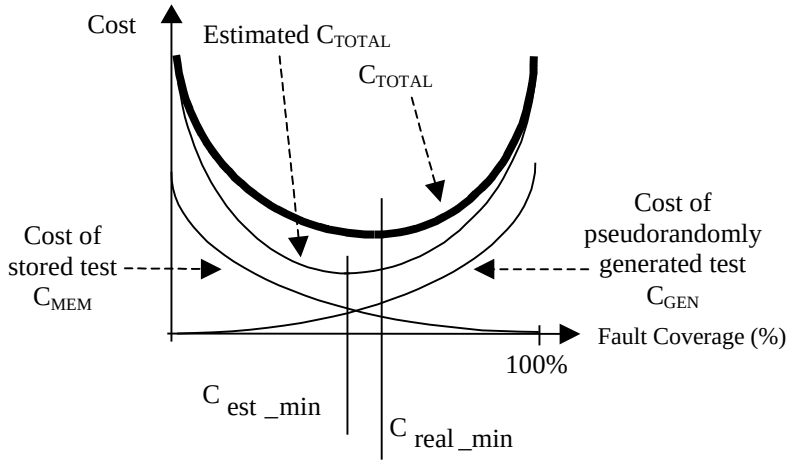


Figure 1. Cost calculation for hybrid BIST

We can define the total test cost of the hybrid BIST C_{TOTAL} as:

$$C_{TOTAL} = C_{GEN} + C_{MEM} = \alpha L + \beta S$$

where C_{GEN} is the cost related to the time for generating L pseudorandom test patterns (number of clock cycles), C_{MEM} is related to the memory cost for storing S precomputed test patterns, and α, β are constants to map the test length and memory space to the costs of the two parts of the test solutions to be mixed.

Figure 1 illustrates also how the cost of pseudorandom test is increasing when striving to higher fault coverage (the C_{GEN} curve). In general, it can be very expensive to achieve high fault coverage with pseudorandom test patterns only. The C_{MEM} curve describes the cost that we have to pay for storing precomputed tests at the given fault coverage reached by pseudorandom testing. The total cost C_{TOTAL} is the sum of the above two costs. The C_{TOTAL} curve is illustrated in Figure 1, where the minimum point is marked as C_{real_min}. The main purpose of this work is to develop a method to find this point quickly. Figure 1 shows also the estimated C_{TOTAL} curve with its minimum at C_{est_min}, which is used for fast estimation of C_{TOTAL} to be discussed in section 3.

In our cost calculations, weights α and β reflect the correlation between the cost and the pseudorandom test time (number of clock cycles used) and between the cost and the memory size needed for storing the precomputed test sequence, respectively. For simplicity we assume here $\alpha = 1$, and $\beta = B$ where B is the number of bytes of the input test vector to be applied on the core under test (CUT). Hence, in the following we will use, as the cost units, number of clocks used for pseudorandom test generation and number of bytes in the memory needed for storing the precomputed test patterns.

Creating the curve C_{GEN} is not difficult. For this purpose, a simulation of the behavior of the LSFR used for pseudorandom test pattern generation is needed. A fault simulation should be carried out for the complete test sequence generated by the LFSR. As a result of such a simulation, we find for each clock cycle the list of faults which were covered at this clock cycle.

In Table 1 a fragment of the results of BIST simulation for the ISCAS'85 circuit c880 [1] is demonstrated. Here

- k is the number of the clock cycle,

k	$r_{DET}(k)$	$r_{NOT}(k)$	$FC(k)$	k	$r_{DET}(k)$	$r_{NOT}(k)$	$FC(k)$
0	155	839	15.593561%	148	13	132	86.720322%
1	76	763	23.239437%	200	18	114	88.531189%
2	65	698	29.778671%	322	13	101	89.839035%
3	90	608	38.832996%	411	31	70	92.957748%
4	44	564	43.259556%	707	24	46	95.372231%
5	39	525	47.183098%	954	18	28	97.183098%
10	104	421	57.645874%	1535	4	24	97.585510%
15	66	355	64.285713%	1560	8	16	98.390343%
20	44	311	68.712273%	2153	11	5	99.496979%
28	42	269	72.937622%	3449	2	3	99.698189%
50	51	218	78.068413%	4519	2	1	99.899399%
70	57	161	83.802818%	4520	1	0	100.000000%
100	16	145	85.412476%				

Table 1. Pseudorandom test results

- $r_{DET}(k)$ is the number of new faults detected (covered) by the test pattern generated at the clock signal k,
- $r_{NOT}(k)$ is the number of faults not yet covered by the sequence of patterns generated by the k clock signals,
- $FC(k)$ is the fault coverage reached by the sequence of patterns generated by the k clock signals

In the list of BIST simulation results not all clock cycles should be presented. We are only interested in the clock numbers at which at least one new fault will be covered, and the total fault coverage for the pseudorandom test sequence up to this clock number increases. Let us call such clock numbers and the corresponding pseudorandom test patterns *resultative clocks* and *resultative patterns*. The rows in Table 1 correspond to the resultative clocks, but not all, for the circuit c880.

If we decide to switch from pseudorandom mode to the deterministic mode after the clock number k, then $L = k$.

More difficult is to find the values for βS. Let $t(k)$ be the number of test patterns needed to cover $r_{NOT}(k)$ not yet detected faults (these patterns should be precomputed and used as stored test patterns in the hybrid BIST). As an example, these data for the circuit c880 are depicted in Table 2. Calculation of the data in the column $t(k)$ of Table 2 is the most expensive procedure. In the following section the difficulties and possible ways to solve the problem are discussed.

k	$t(k)$	k	$t(k)$
0	104	148	46
1	104	200	41
2	100	322	35
3	101	411	26
4	99	707	17
5	99	954	12
10	95	1535	11
15	92	1560	7
20	87	2153	3
28	81	3449	2
50	74	4519	1
70	58	4520	0
100	52		

Table 2. ATPG results

3. Optimization algorithm

There are two approaches to find *t(k)*: ATPG based and fault table based. Let us have the following notations:

- *i – the current number of the entry in the tables for PRG and ATPG;*
- *k(i) – the number of the clock cycle of the resultative clock i;*
- $R_{DET}(i)$ *- the set of new faults detected (covered) by the pseudorandom test pattern which is generated at the resultative clock signal number i;*
- $R_{NOT}(i)$ *- the set of not yet covered faults after applying the pseudorandom test pattern number i;*
- *T(i) - the set of test patterns needed and found by the ATPG to cover the faults in* $R_{NOT}(i)$*;*
- *N – the number of all resultative patterns in the sequence created by the pseudorandom test;*
- *FT – the fault table for a given set of tests T and for the given set of faults R: the element* ε_{jk} *(j is ranged over the test set T, and k over the fault set R) in the table is defined as follows.* $\varepsilon_{jk} = 1$ *if the test* $t_j \in T$ *detects the* $r_k \in R$*; otherwise* $\varepsilon_{jk} = 0$*.*

Algorithm1: ATPG based approach for finding test sets *T(i)*

1. Let $q := N$;
2. Generate for $R_{NOT}(q)$ a test set $T(q)$, $T := T(q)$, $t(q) := |T(q)|$;
3. For all $q = N-1, N-2, \ldots 1$:
   Generate for the faults $R_{NOT}(q)$ not covered by test T a test set $T(q)$, $T := T + T(q)$, $t(q) := |T|$.

End.

Algorithm 2: Fault Table based approach for finding test sets *T(k)*

1. Let $q = 1$; calculate the test $T(q)$ for the whole set of faults R, create the fault table FT;
2. For all $q = 2,3, \ldots N$:
   Create a new fault table FT by removing from it the faults $R_{DET}(q-1)$, and optimize the test set $T(q-1)$ in relation to the new FT. The optimized test set is $T(q)$.

End.

For the first approach, the whole experiment of simulating the pseudorandom generation (PRG) behavior and of finding the numbers of test patterns to be stored for all possible switching points from PRG to stored test patterns was carried out for the whole set of ISCAS'85 benchmark circuits within 8 hours. The data of these experiments are depicted in Table 3.

In the case of very large circuits both of these algorithms will lead to very expensive and time-consuming experiments. For such situations we have developed an estimation algorithm to search the optimum solution by using just a few samples from the whole test generation experiments set. As available data for such kind of estimation, the number of not yet covered faults in $R_{NOT}(k)$ can be used. The value of $R_{NOT}(k)$ can be acquired directly

from the PRG simulation results and is available for every significant time moment (see Table 1). Based on the value of $|R_{NOT}(k)|$ it is possible to reason about the expected number of test patterns needed for covering the faults in $R_{NOT}(k)$. The starting point of the search procedure may be found by giving rough estimation of the total cost based on the value of $|R_{NOT}(k)|$.

Estimated cost would suggest the first solution for dividing the BIST into two parts (PRG and deterministic based). Based on this starting point a search for the optimum can be carried out by using some classically known methods, like Tabu search [2], to search for the global optimum. The results of the estimation algorithm are presented in Table 3. The cost curves used for optimization of hybrid BIST for some ISCAS'85 circuits are graphically depicted in Figure 2.

4. Experimental results

Experiments were carried out on the ISCAS'85 benchmark circuits for investigating the efficiency of the method for optimizing the hybrid BIST based on the first algorithm of calculating the test sets for each possible point of switching from PRG mode to the stored deterministic patterns mode. The experiments were carried out using Turbo Tester toolset [5], [9] and the results are presented in Table 3.

Benchmark circuits	c432	c499	c880	c1355	c1908
1. Simulated clocks	470	1437	4520	1371	1551
2. Resultative clocks	77	110	120	111	163
3. PRG fault coverage	86.36	81.19	86.22	85.53	72.17
4. Remained faults	84	226	137	234	482
5. Total fault coverage	93.02	99.33	100	99.51	97.40
6. L: PRG test length	91	78	121	121	105
7. M: Stored test length	21	60	48	52	123
8. Number of inputs	36	41	60	41	33
9. Total cost	186	386	481	388	612
10.Estim. PRG length	107	106	168	429	400
11.Estimated cost	188	396	491	536	732
12.Estimation accuracy	0.99	0.97	0.98	0.62	0.80
Benchmark circuits	c2670	c3540	c5315	c6288	c7552
1. Simulated clocks	8455	3113	1455	31	19029
2. Resultative clocks	117	258	215	25	342
3. PRG fault coverage	84.46	94.81	98.76	99.34	92.44
4. Remained faults	427	521	83	51	673
5. Total fault coverage	95.51	95.44	98.89	99.34	97.69
6. L: PRG test length	444	297	711	31	583
7. M: Stored test length	77	110	12	0	61
8. Number of inputs	233	50	178	32	207
9. Total cost	2687	889	985	31	2161
10. Estim. PRG length	419	682	515	19	790
11. Estimated cost	2718	1041	1095	35	2266
12. Estimation accuracy	0.99	0.83	0.89	0.87	0.95

Table 3. Experimental results of creating optimized hybrid BIST

In rows 1 and 2, respectively, the maximum numbers of clocks simulated for the PRG mode, and the number of *resultative* clocks are given. For each resultative clock the parameters $r_{DET}(i)$, $r_{NOT}(i)$ and $FC(i)$ were calculated (as in Table 1).

On the basis of these data the tests were generated for all sets of faults $R_{DET}(i)$, i = 1,2,..,N, according to Algorithm 1. Then, the costs $C_{GEN} = \alpha L$, $C_{MEM} = \beta S$ and $C_{TOTAL} = C_{GEN} + C_{MEM}$ were calculated as functions from the number of clocks to be used in the PRG mode of the hybrid BIST.

The switching point from the PRG mode to the stored deterministic patterns mode was found at the minimum of C_{TOTAL}. This solution is characterized in the rows 3-9 as the fault coverage reached in the PRG mode (3), the number of faults to be tested in the stored patterns mode (4), total fault coverage reached by both modes together (5), the length L of the pseudorandom test (6), the length M of the stored test (7), the number of inputs of the circuit needed for calculation the memory weight β in the cost function (8), and the total cost of the chosen solution (9).

An attempt was made to find a fast cost estimation function based on the numbers of remaining faults for all resultative clocks (as possible switching points from PRG mode to the stored patterns mode). Based on statistical analysis of the costs calculated, the following coefficient was chosen: 1 remaining fault = 0,45 test patterns needed to cover it. By this coefficient, a new cost calculation function was achieved:

$$C_{TOTAL_EST} = C_{GEN} + C_{MEM_EST} = \alpha L + 0.45\beta F,$$

where F is the number of remained faults not covered by the PRG mode. The values of minimum of C_{TOTAL_EST} for each benchmark circuit are given in the row 11. The estimated switching point from the PRG mode to the stored patterns mode was found at the minimum of C_{TOTAL_EST} (10). The accuracy of the estimation is depicted in the row 12.

The accuracy of the estimation of the cost depends of how close are the curves C_{MEM_EST} and C_{MEM} to each other, more exactly from the function $C_{MEM_EST} = f(r_{NOT}(k))$. In this work, based on the statistical analysis of experimental data, for simplicity an average coefficient was chosen $f(r_{NOT}(k)) \equiv 0{,}45$. How close the estimated cost curves and the real cost curves can be is observable in Figure 2 for some ISCAS'85 benchmark circuits. Different from Figure 1 the costs are presented here as functions of clock numbers (not of the fault coverage). In this case the curve αL has a linear form, and it is easier to follow the relationships between the memory cost function and the total cost function.

In general it seems very difficult to find a good approximation function $f(r_{NOT}(k))$ general enough for different circuits. However we have developed an approach to suggest a good point for starting a search for the global optimum by sampled calculation of the real cost function C_{MEM}.

5. Conclusions and future work

This paper describes a hybrid BIST solution for testing systems-on-chip, which combines pseudorandom test patterns with deterministic test patterns. For selecting the optimal switching moment from PRG mode to the stored deterministic patterns mode, two accurate algorithms together with one fast estimation algorithm are given. The accurate algorithms are rather time consuming and cannot be effectively used for very complex circuits. For finding the optimal hybrid BIST solution for all 10 ISCAS'85 benchmark circuits, 8 hours of computing time was needed.

As it was shown by experimental results, the fast cost estimation algorithm can give results with reasonable accuracy and can speed up the optimization process significantly. In general, the proposed estimation method can be used as a good starting point for the search for a global optimum by few additional exact calculations of the real cost.

Experimental results have shown that the proposed approach is feasible and efficient for finding optimized solutions for hybrid BIST architectures in systems-on-chip.

As a future work we would like to investigate possibilities to use the proposed approach for parallel testing issue (testing multiple cores at the same time) and to use the same ideas for sequential cores.

Acknowledgements

The authors appreciate the work of Jaan Raik, Priidu Paomets, Elmet Orasson and Artur Jutman from Tallinn Technical University for developing software tools, and carrying out the experiments.

6. References

[1] F. Brglez, H. Fujiwara. "A Neutral Netlist of 10 Combinational Benchmark Circuits and a Target Translator in Fortran," IEEE Int. Symp. on Circuits and Systems, pp. 663-698, June 1985.

[2] F. Glover, "Future paths for integer programming and links to artificial intelligence," Computers & Ops. Res., pp. 533-549, No. 5, 1986

[3] S. Hellebrand, S. Tarnick, J. Rajski, B. Courtois, "Generation Of Vector Patterns Through Reseeding of Multiple-Polynomial Linear Feedback Shift Registers," IEEE Int. Test Conference (ITC'92), pp. 120-129, Baltimore, 1992

[4] S. Hellebrand, H.-J. Wunderlich, A. Hertwig, "Mixed-Mode BIST Using Embedded Processors," Journal of Electronic Testing: Theory and Applications," pp. 127-138, No. 12, 1998

[5] G. Jervan , A. Markus, P. Paomets, J. Raik, R. Ubar. "A CAD system for Teaching Digital Test," 2nd European Workshop on Microelectronics Education, pp. 287-290, Noordwijkerhout, the Netherlands, May 14-15, 1998

[6] E. J. Marinissen, Y. Zorian, "Challenges in Testing Core-Based System ICs," IEEE Communications Magazine, pp. 104-109, June 1999.

[7] "The National Technology Roadmap for Semiconductors," Semiconductor Industry Assoc., San Jose, Calif., 1997, 1998

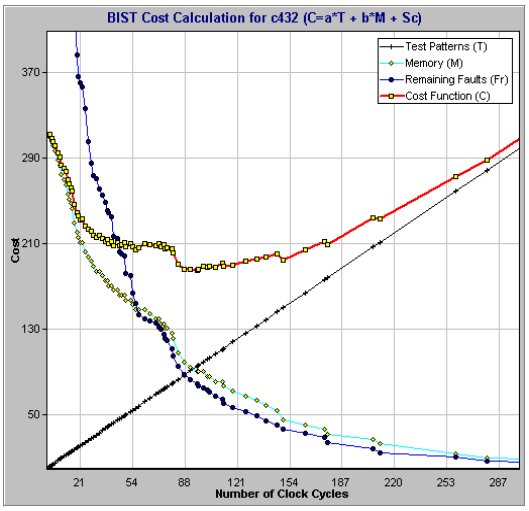
[8] M. Sugihara, H. Date, H. Yasuura, "Analysis and Minimization of Test Time in a Combined BIST and External Test Approach," Design, Automation & Test In Europe Conference (DATE 2000), pp. 134-140, Paris, France, March 2000

[9] Turbo Tester Reference Manual. Version 3.99.03. Tallinn Technical University 1999. http://www.pld.ttu.ee/tt

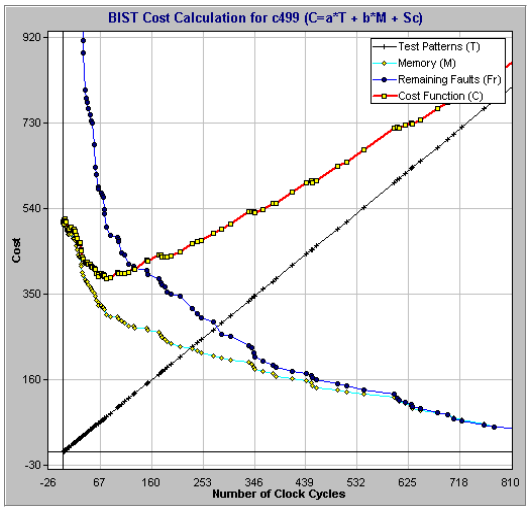
[10] "Virtual Socket Interface Architectural Document," VSI Alliance, Nov. 1996

[11] N. Zacharia, J. Rajski, J. Tyzer, "Decompression of Test Data Using Variable-Length Seed LFSRs," 13th VLSI Test Symposium, pp. 426-433, 1995.

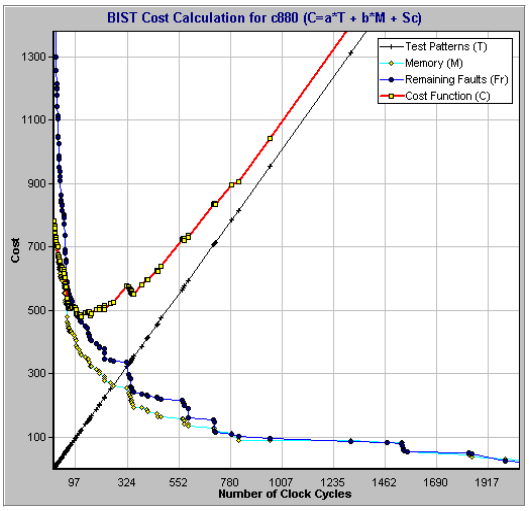
[12] Y. Zorian, E. J. Marinissen, S. Dey, "Testing Embedded Core-Based System Chips," IEEE International Test Conference (ITC), pp. 130-143, Washington, DC, October 1998. IEEE Computer Society Press.



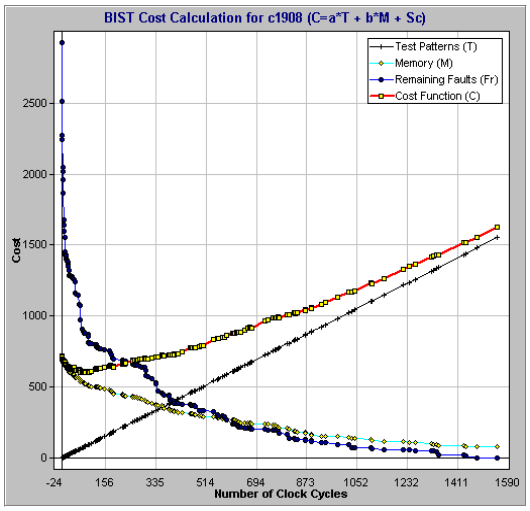
C432



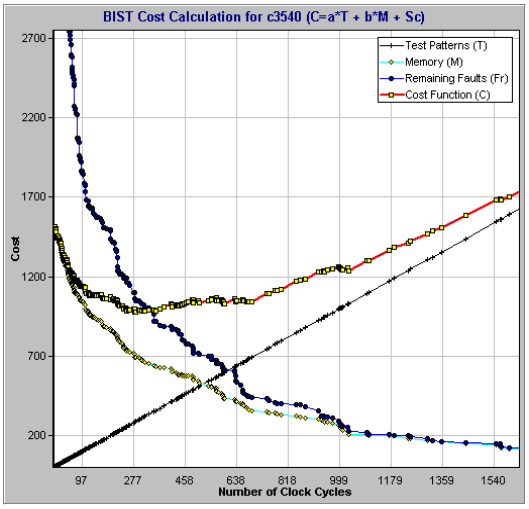
C499



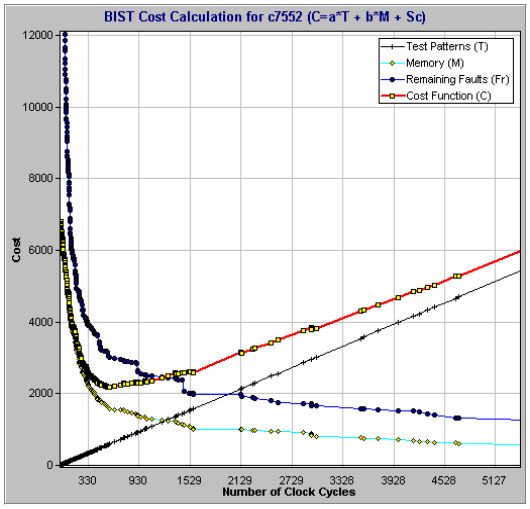
C880



C1908



C3540



C7552

Figure 2. Cost curves of hybrid BIST for ISCAS'85 benchmark circuits