Fast Test Cost Calculation for Hybrid BIST in Digital Systems[1]

Elmet Orasson, Rein Raidma, Raimund Ubar
Tallinn Technical University
EE-12618 Tallinn, Estonia
elmet@pld.ttu.ee, b970843@ttu.ee, raiub@pld.ttu.ee

Gert Jervan, Zebo Peng
Linköping University
SE-581 83 Linköping, Sweden
gerje@ida.liu.se, zebpe@ida.liu.se

Abstract

This paper presents a hybrid BIST solution for testing systems-on-chip which combines pseudorandom test patterns with stored precomputed deterministic test patterns. A procedure is proposed for fast calculation of the cost of hybrid BIST at different lengths of pseudorandom test to find an optimal balance between test sets, and to perform core test with minimum cost of both, time and memory, and without losing in test quality. Compared to the previous approach, based on iterative use of deterministic ATPG for evaluating the cost of stored patterns, in this paper a new, extremely fast procedure is proposed, which calculates costs on a basis of fault table manipulations. Experiments on the ISCAS benchmark circuits show that the new procedure is about two orders of magnitude faster than the previous one.

1. Introduction

To test the electronic system we need test pattern source and sink together with an appropriate test access mechanism (TAM) [1]. Such a test architecture can be implemented 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 rapid advances in recent years have enabled the integrated circuits (ICs) manufactures to move towards very deep submicron technologies and to integrate several complex functional blocks into one single chip. The internal speed of such a Systems-on-Chip (SoC) is constantly increasing but the technology used in ATE is always one step behind and therefore the ATE solution is already unacceptably expensive and inaccurate [2]. Therefore, in order to apply at-speed tests and to keep the test 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 consists of a test pattern generator (TPG), a test response analyzer (TRA) and a BIST control unit (BCU), all implemented on the chip. This approach supports at-speed tests and eliminates the need for an external tester. Different BIST approaches have been available for a while and have got wide acceptance especially for memory test. For logic BIST (LBIST) there is still no industry-wide acceptance. One of the main reasons is the hardware overhead required to implement a BIST architecture. The BIST approach can also introduce additional delay to the circuitry and requires a relatively long test application time. Nevertheless, even LBIST is becoming increasingly popular, since BIST is basically the only practical solution to perform at-speed test, and can be used not only for manufacturing test but also for periodical field maintenance tests.

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

The main concern of many existing 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. In [8], a method for optimizing the cost of hybrid BIST is proposed which is based on iterative use of deterministic ATPG to evaluate different combinations of pseudorandom and deterministic test patterns. Since repeated runs of ATPG are very expensive, a method was proposed to speed up the optimization procedure by using approximate estimations of the expected cost for different possible solutions.

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

The main objective of the current work is to propose a solution to find the optimal balance between the on-line pseudorandom test pattern generation and usage of stored precomputed 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 calculate, with very low computational time, a complete hybrid test set, and to derive from it the optimal time-moment to stop pseudorandom test generation and to apply deterministic patterns.

A similar problem has been addressed in [6], where an approach to minimize testing time has been presented. The authors have shown that hybrid BIST (or Combination of BIST and External Test, CBET, in their terminology) can achieve shorter testing time than pure pseudorandom test or pure externally applied deterministic test. The authors have made a realistic assumption that externally applied test is much slower than LFSR generated one and therefore internally generated test vectors should be used as much as possible. However, the proposed algorithm is not addressing total cost minimization (time and memory) and is therefore only a special case of our approach.

The rest of the paper is organized as follows. In section 2 we discuss the concepts of hybrid BIST. Section 3 introduces the algorithm to calculate the optimal hybrid BIST set. In section 4 we present the experimental results which demonstrate the feasibility and efficiency of our approach. In section 5 we will draw some conclusions together with an introduction to future work.

2. Cost Factors for Hybrid BIST

As test patterns, generated by LFSRs, are pseudorandom by nature and have linear dependencies, 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 in order to speed up the testing process, we have to apply deterministic test patterns targeting the random resistant and difficult to test faults. This hybrid BIST approach starts with on-line generation of pseudorandom test sequence with a length of L. On the next stage, stored test approach takes place. For the stored approach, precomputed test patterns, stored in the memory, are applied to the core under test to reach 100% fault coverage. For off-line generation of S deterministic test patterns (the number of stored test patterns) arbitrary software test generators may be used, based on deterministic, random or genetic algorithms.

In hybrid BIST, 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 however 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. A situation is illustrated, where 100% fault coverage is achievable with pseudorandom vectors only, although it takes enormously long time to do it. In the case of large and complex designs 100% fault coverage is not always achievable, however.

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

$$C_{TOTAL} = C_{GEN} + C_{MEM}$$

where C_{GEN} is the cost related to the time for generating the pseudorandom test patterns (number of

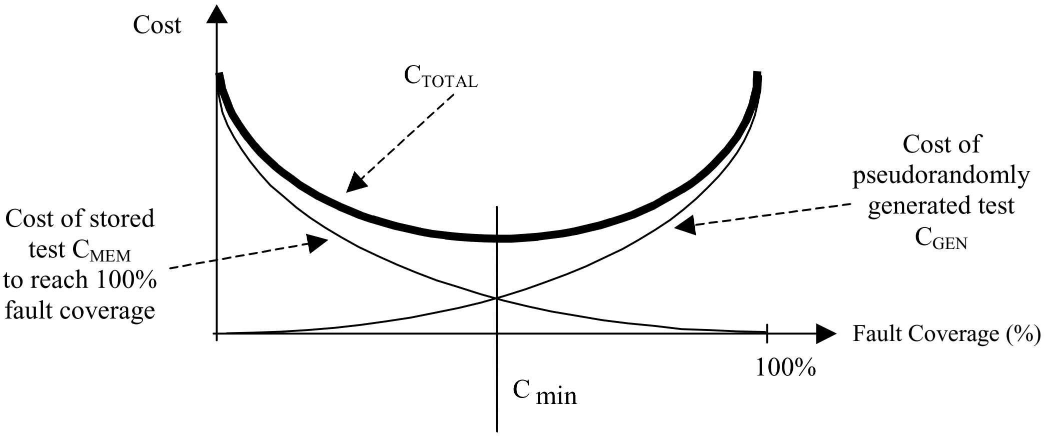

Figure 1: Cost calculation for hybrid BIST (with 100% assumption)

clock cycles), C_{MEM} is related to the memory cost for storing the precomputed test patterns to improve the pseudorandom test set.

Figure 1 illustrates also how the cost of pseudorandom test increases when striving for 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 additional precomputed tests from the fault coverage level reached by pseudorandom testing to 100%. 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_{min}. The main purpose of this work is to find a fast method for calculating the curve C_{TOTAL} to find the minimal cost C_{min}.

As mentioned, in many cases 100% fault coverage is not achievable with pseudorandom vectors only. Therefore we have to include this assumption to the total cost calculation and the new situation is illustrated in Figure 2, where the horizontal axis indicates the number of pseudorandom test patterns applied, instead of fault coverage level. The curve for the total cost C_{TOTAL} is still the sum of two cost curves C_{GEN} + C_{MEM} with the new assumption that the maximum fault coverage reachable by only deterministic ATPG is achieved by the hybrid BIST.

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

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

where L is the length of the pseudorandom test sequence; S is the number of deterministic patterns; and 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 deterministic test sequence, respectively. For simplicity, we assume here that $\alpha = 1$, and $\beta = B$ where B is the number of bytes of an input test vector to be applied to the CUT. Hence, in the following we will use, as the cost units, number of cycles used for pseudorandom test generation and number of bytes in the memory needed for storing the precomputed test patterns.

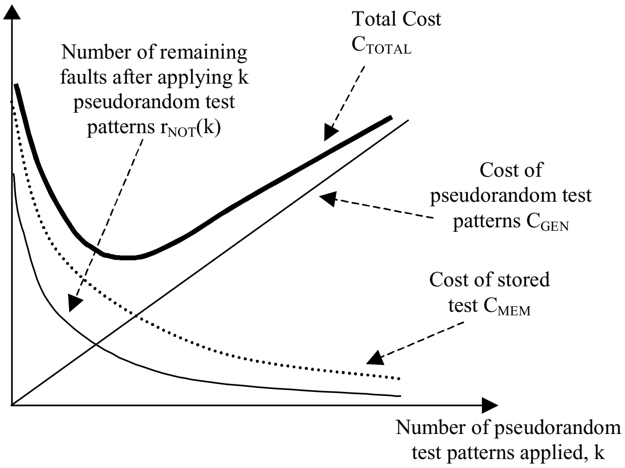

Figure 2: Cost calculation for hybrid BIST

In Table 1 a fragment of the results of BIST simulation for the ISCAS'85 circuit c880 [9] is given, where

- *k* is the clock counter;
- $r_{DET}(k)$ is the number of new faults detected by the group of test patterns generated between the last and the current entry at k in the table;
- $r_{NOT}(k)$ is the number of faults not yet covered by the sequence of patterns generated during all the k clock cycles; and
- *FC(k)* is the fault coverage reached by the sequence of patterns generated during all the k clock cycles.

In the list of BIST simulation results not all clock cycles are presented. We are only interested in the clock numbers at which at least one new fault will be covered, and thus the total fault coverage for the pseudorandom test sequence up to this clock number increases. Let us

k	$r_{DET}(k)$	$r_{NOT}(k)$	*FC(k)*
1	155	839	15.593561%
2	76	763	23.239437%
3	65	698	29.778671%
4	90	608	38.832996%
5	44	564	43.259556%
6	39	525	47.183098%
10	104	421	57.645874%
16	66	355	64.285713%
20	44	311	68.712273%
29	42	269	72.937622%
50	51	218	78.068413%
70	57	161	83.802818%
100	16	145	85.412476%
149	13	132	86.720322%
200	18	114	88.531189%
323	13	101	89.839035%
412	31	70	92.957748%
708	24	46	95.372231%
955	18	28	97.183098%
1536	4	24	97.585510%
1561	8	16	98.390343%
2154	11	5	99.496979%
3450	2	3	99.698189%
4520	2	1	99.899399%
4521	1	0	100.000000%

Table 1. Pseudorandom test results

call such clock numbers and the corresponding pseudorandom test patterns *resultative clocks* and *resultative patterns*. The rows in Table 1 represent the resultative clocks, but not all (we only give some resultative points for illustrative purpose), for the circuit c880.

If we decide to switch from the on-line pseudorandom test generation mode to the deterministic stored pattern mode after the clock number *k*, then $L = k$.

Creating the curves C_{GEN} and $r_{NOT}(k)$ 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 up to this clock cycle. By removing these faults from the complete fault list, we will know the number of faults remaining to be tested.

More difficult is to find the values of βS, the cost for storing additional deterministic patterns in order to reach the given fault coverage level (100% in the ideal case). 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).

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 [8] for each value of *k*, ATPG was used to generate the test patterns for testing not yet detected faults.

Since usage of ATPG is a very costly procedure, we present in the following a new algorithm based on iterative transformations of fault tables. The new algorithm allows dramatic reduction of computation time in the hybrid BIST cost calculations.

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. Fast procedure for calculating stored test patterns

Calculation of the data in the column $t(k)$ of Table 2 is the most expensive procedure. In the following section a procedure for fast calculation of this information is presented. Let us have the following notations:

- N is the number of all resultative patterns in the sequence created by the pseudorandom test i.e., the number of entries in the tables for PRG and ATPG (see Tables 1 and 2)
- i is the current number of the entry in the tables (i =1, 2, …, N)
- k_i is the total number of the clock cycles of the pseudorandom test sequence up to the entry i
- $R_{DET,i}$ is the set of faults detected (covered) by the pseudorandom test pattern sequence which is generated at the resultative clock signal number k_i
- $R_{NOT,i}$ is the set of not yet detected (covered) faults after applying the pseudorandom test pattern sequence at the resultative clock signal number k_i
- T_i is the set of stored test patterns generated by a deterministic ATPG to cover the faults in $R_{NOT,i}$.

The fault table FT for a general case is defined as follows: given a set of test patterns $T = \{t_i\}$ and a set of faults $R = \{r_j\}$, $FT = \| \varepsilon_{ij} \|$ where $\varepsilon_{ij} = 1$ if the test $t_i \in T$ detects the $r_j \in R$, and $\varepsilon_{ij} = 0$ in the opposite case. We denote by $R(t_i) \subset R$ the subset of faults detected by the test pattern $t_i \in T$.

We start the procedure for a given circuit by generating a test set *T* which gives the 100% (or as high as possible) fault coverage. This test set can be served as a stored test if no on-line generated pseudorandom test sequence will be used. By fault simulation of the test set *T* for the given set of faults *R* of the circuit, we create the fault table *FT*.

Suppose now, that we use a pseudorandom test sequence T^L with a length *L* which detects a subset of faults $R^L \subset R$. It is obvious that when switching from the pseudorandom test mode with a test set T^L to the precomputed stored test mode with a *T*, the test set *T* can be significantly reduced. At first, by the fault subtraction operation $R(t_i) - R^L$ we can update all the contributions of the test patterns t_i in *FT* (i.e. to calculate for all t_i the remaining faults they can detect after performing the pseudorandom test). After that we can use any procedure of static test compaction to minimize the test set *T*.

The described procedure of updating the fault table *FT* can be carried out iteratively for all possible breakpoints $i = 1, 2, \dots, N$ of the pseudorandom test sequence by the following algorithm.

Algorithm 1:

```
1. Calculate the whole test $T = \{t_j\}$ for the whole set of faults $R = \{r_j\}$ by any ATPG to reach as high fault coverage $C$ as possible
2. Create for $T$ and $R$ the fault table $FT = \{R(t_j)\}$
3. Take $i = 1$; Rename: $T_i = T, R_i = R, FT_i = FT$
4. Take $i = i + 1$
5. Calculate by fault simulation the fault set $R_{DET,i}$
6. Update the fault table: $\forall j, t_j \in T_i$: $R(t_j)$ - $R_{DET,i}$
7. Remove from the test set $T_i$ all the test patterns $t_j \in T_i$ where $R(t_j) = \emptyset$
8. Optimize the test set $T_i$ by any test compaction algorithm; fix the value of $S_i = | T_i |$ as the length of the stored test for $L = i$;
9. If $i < L$, go to 4;
End.
```

It is easy to understand that for each value $L = i$ (the length of the pseudorandom test sequence) the procedure guarantees the constant fault coverage C of the hybrid BIST. The statement comes from the fact that the subset T_i of stored test patterns is complementing the pseudorandom test sequence for each $i = 1, 2, \ldots, N$ to reach the same fault coverage reached by T.

As the result of *Algorithm 1*, the numbers of precomputed deterministic test patterns $S_i = |T_i|$ to be stored and the subsets of these patterns T_i for each $i = 1, 2, \ldots, N$ are calculated. On the basis of this data the cost of stored test patterns for each i can be calculated by the formula $C_{MEM} = \beta S_i$. From the curve of the total cost $C_{TOTAL}(i) = \alpha L + \beta S$ the value of the minimum cost of the hybrid BIST *min* $\{C_{TOTAL}(i)\}$ can be easily found.

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 and for comparing with the previous algorithm used in [8]. Experiments were carried out using Turbo Tester toolset [10], [11] for deterministic test pattern generation and fault simulation, and using the test compaction tool [12]. The results are presented in Table 3.

In the columns of Table 3 the following data is depicted: ISCAS'85 benchmark circuit name, L_{PR} - length of the pseudorandom test sequence, L_{DET} - the number of test patterns generated by the deterministic ATPG, C_{PR} - the fault coverage of the pseudorandom test sequence, C_{DET} - the total fault coverage of the hybrid BIST (after applying deterministic test patterns), N - number of all resultative patterns in the pseudorandom test sequence, T_G - the time (sec) needed for ATPG to generate the deterministic test set, T_A - the time (sec) needed for carrying out manipulations on fault tables (subtracting faults, and compacting the test set), T_{OLD} - the time (sec) needed for calculating the cost curve for hybrid BIST by the previous method [8], T_{NEW} - the time (sec) needed for calculating the cost curve for hybrid BIST by the new proposed method, the advantage of the proposed method compared to the previous one as the relation T_{OLD}/T_{NEW}. The total testing time for both methods were calculated as follows:

$$T_{OLD} = N * T_G$$

$$T_{NEW} = T_G + N * T_A.$$

In fact, the values for T_G and T_A differ for the different values of $i = 1, 2, \ldots, N$. However the differences were in the range of few percents, which allowed us to neglect this impact and to use the average values of T_G and T_A.

Circuit	L_{PR}	L_{DET}	C_{PR}	C_{DET}	N	T_G	T_A	T_{OLD}	T_{NEW}	T_O/T_N
C432	780	80	93.02	93.02	81	20.10	0.01	1632.9	21.0	**77.75**
C499	2036	132	99.33	99.33	114	0.65	0.02	74.1	2.9	**25.55**
C880	5589	77	100.00	100.00	114	0.15	0.02	17.1	2.4	**7.13**
C1355	1522	126	99.51	99.51	109	1.22	0.03	133.0	4.5	**29.56**
C1908	5803	143	99.48	99.48	183	11.65	0.07	2132.0	24.5	**87.02**
C2670	6581	155	84.92	99.51	118	1.95	0.09	230.1	12.6	**18.25**
C3540	8734	211	95.54	95.54	265	85.29	0.14	22601.9	122.4	**184.66**
C5315	2318	171	98.89	98.89	252	10.29	0.11	2593.1	38.0	**68.24**
C6288	210	45	99.34	99.34	53	3.79	0.04	200.9	5.9	**34.05**
C7552	18704	267	93.67	97.14	279	53.78	0.27	15004.6	129.1	**116.22**

Table 3. Experimental results. Comparison of two algorithms

Circuit	L_{MAX}	L_{OPT}	S_{MAX}	S_{OPT}	B_k	C_{TOTAL}
C432	780	91	80	21	4	186
C499	2036	78	132	60	6	386
C880	5589	121	77	48	8	481
C1355	1522	121	126	52	6	388
C1908	5803	105	143	123	5	612
C2670	6581	444	155	77	30	26867
C3540	8734	297	211	110	7	889
C5315	2318	711	171	12	23	985
C6288	210	20	45	20	4	100
C7552	18704	583	267	61	51	2161

Table 4. Experimental results. Parameters for optimized hybrid BIST

The switching point from the PRG mode to the stored deterministic patterns mode was found at the minimum of C_{TOTAL}. In Table 4 the parameters of optimized hybrid BIST are depicted: ISCAS'85 benchmark circuit name, L_{MAX} - the maximum length of the simulated pseudorandom test sequence, L_{OPT} - the length of the pseudorandom test sequence for the optimized BIST, S_{MAX} - the maximum number of test patterns generated by the deterministic ATPG, S_{OPT} - the number of stored test patterns for the optimized BIST, B_k - the number of bytes needed for storing the input test pattern for the circuit *k,* and C_{TOTAL} - the total cost of the optimized hybrid BIST, calculated by the formula $C_{TOTAL} = L_{OPT} + B_k * S_{OPT}$.

In Figure 3, the curves of the cost $C_{GEN} = L$ (denoted on Fig.3 as *T*) for on-line pseudorandom test generation, the cost $C_{MEM} = B_k*S$ (denoted as *M*) for storing the test patterns, the number $|R_{NOT}|$ of not detected faults after applying the pseudorandom test sequence (denoted as F_r), and the total cost function C_{TOTAL} are depicted for selected benchmark circuits C432, C499, C880, C1908, C3540 and C7552 (Sc = 0 is used as a constant in the cost function formula).

5. Conclusions and future work

This paper describes a hybrid BIST solution for testing systems-on-chip. It combines on-line pseudorandom test pattern generation with using precomputed and stored deterministic test patterns. For selecting the optimal switching moment from pseudorandom patterns mode to the stored deterministic patterns mode, a fast algorithms to calculate the total cost is given.

As it was shown by experimental results, the new cost calculation algorithm calculates the total cost of the hybrid BIST much faster (7 - 184 times faster) than previously proposed algorithm and can therefore speed up the total cost minimization process significantly.

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 in case of sequential cores.

6. Acknowledgements

The authors appreciate the work of Jaan Raik from Tallinn Technical University for developing the software tools used for the experimental work, and providing support for the experiments.

7. References

[1] 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.

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

[3] M. Chatterjee, D. K. Pradhan, "A novel pattern generator for near-perfect fault-coverage," *VLSI Test Symposium*, pp. 417-425, 1995

[4] 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

[5] 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

[6] 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

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

[8] G. Jervan, Z. Peng, R. Ubar, "Test Cost Minimization for Hybrid BIST." *The IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems*, pp.283-291, Yamanashi, Japan, October 2000.

[9] F. Brglez, P. Pownall, R. Hum. "Accelerated ATPG and fault grading via testability analysis," *IEEE Int. Symp. on Circuits and Systems*, pp. 663-698, June 1985.

[10] 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.

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

[12] A.Markus, J.Raik, R.Ubar, "Test Set Minimization Using Bipartite Graphs," *Proc. of the 6th Baltic Electronics Conference,* pp. 175-178, Tallinn, Estonia, October 1998.

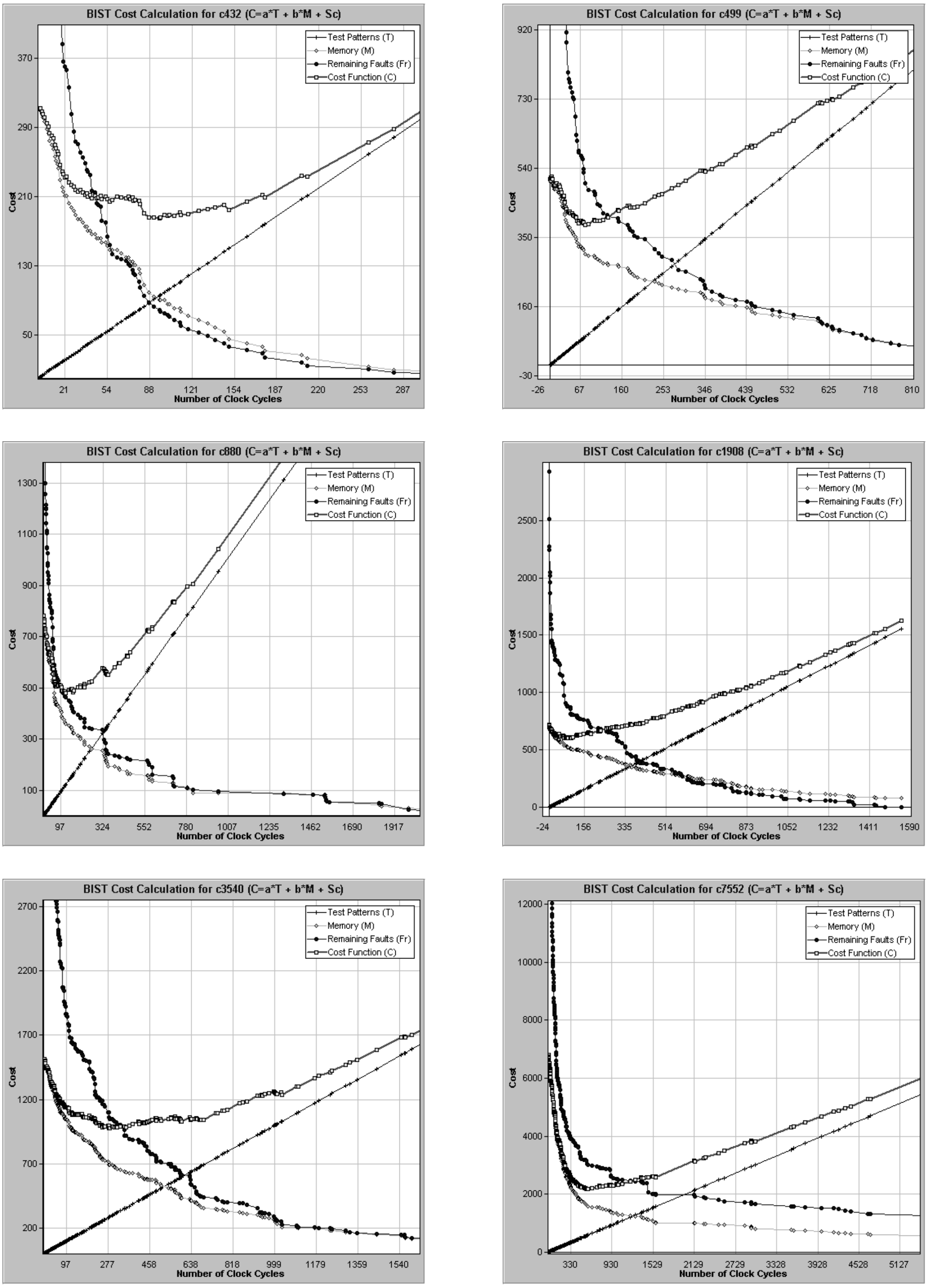

Figure 3. Cost curves of hybrid for ISCAS’85 benchmark circuits