

Static Compaction Techniques to Control Scan Vector Power Dissipation

Ranganathan Sankaralingam, Rama Rao Oruganti, and Nur A. Toubia

Computer Engineering Research Center
Department of Electrical and Computer Engineering
University of Texas, Austin, TX 78712-1084
E-mail: {ranga, oruganti, toubia}@ece.utexas.edu

Abstract

Excessive switching activity during scan testing can cause average power dissipation and peak power during test to be much higher than during normal operation. This can cause problems both with heat dissipation and with current spikes. Compacting scan vectors greatly increases the power dissipation for the vectors (generally the power becomes several times greater). The compacted scan vectors often can exceed the power constraints and hence cannot be used. It is shown here that by carefully selecting the order in which pairs of test cubes are merged during static compaction, both average power and peak power for the final test set can be greatly reduced. A static compaction procedure is presented that can be used to find a minimal set of scan vectors that satisfies constraints on both average power and peak power. The proposed approach is simple yet effective and can be easily implemented in the conventional test vector generation flow used in industry today.

1. Introduction

Power dissipation in CMOS circuits is proportional to the amount of switching that takes place [Devadas 95]. Switching activity in an integrated circuit can be much greater during test than during normal circuit operation. During normal operation, typically a relatively small percentage of the flip-flops change value in each clock cycle. However, when scanning in test vectors, typically a much larger percentage of the flip-flops will change value in each clock cycle. Consider the simple case of scanning a vector of alternating ones and zeros (1010...10) into a scan chain. When the vector is scanned in, all of the flip-flops in the scan chain would simultaneously switch resulting in a large current spike. The excessive switching activity during scan testing can cause average power dissipation and peak power during test to be significantly higher than during normal operation. This can cause problems both with heat dissipation and with current spikes.

The amount of heat that can be safely dissipated without damaging the chip is limited by the chip's

package. This must be taken into consideration when scheduling tests and limits either the number of modules that can be tested simultaneously [Zorian 93] or the speed at which the tests can be applied. This in turn impacts test time and test cost.

A chip's power supply/ground pins and distribution system may be designed for handling the peak power that occurs during normal operation, and it may not be able to handle the large peak power that could occur during test. If the peak power during test is too large, then there will be a V_{dd} drop/ground bounce that may cause problems (e.g., memory elements to lose their state or phase-locked loop (PLL) to malfunction). So peak power is another constraint that impacts scan testing.

With the proliferation of portable battery operated devices and the drive toward low power design and low-cost light packages, the power issues for test are becoming increasingly important [Crouch 99]. Techniques for minimizing both average power and peak power during test to satisfy the constraints imposed by the chip package and power supply/ground distribution system are needed.

Researchers have begun looking at ways to control power dissipation during test. Test scheduling algorithms that satisfy power constraints were presented in [Chou 94]. Low power BIST techniques were presented in [Wang 97a], [Hertwih 98], [Wang 99], [Gerstendörfer 99], [Girard 99]. Techniques for minimizing power dissipation when testing combinational circuits were presented in [Wang 94], [Dabholkar 98]. Techniques for minimizing power dissipation during scan testing were presented in [Wang 97b], [Dabholkar 98].

This paper focuses on the problem of minimizing power dissipation during scan testing. Wang and Gupta [Wang 97b] proposed a special ATPG (automatic test pattern generation) procedure that tries to find a set of scan vectors for a circuit that will minimize average power during scan testing. They modify the controllability and observability cost functions in PODEM [Goel 81]. Dabholkar, *et al.*, [Dabholkar 98], proposed heuristic algorithms for test vector ordering and scan chain ordering that try to minimize average power

during scan testing. In this paper, we present a static compaction procedure that compacts a set of test cubes (i.e., test vectors where the unspecified values have been left as X's) in a way that minimizes either average power or peak power. Our procedure can be used to find a minimal set of scan vectors that satisfies constraints on both average power and peak power. The proposed approach is simple and can be easily implemented in the conventional test vector generation flow used in industry today. It will work with any ATPG procedure (including the one described in [Wang 97b]). It is also compatible with the techniques described in [Dabholkar 98].

The motivation behind this research is the following: It has been observed in industry that compacting scan vectors greatly increases the power dissipation for the vectors (generally the power becomes several times greater). The compacted scan vectors often can exceed the power constraints and hence cannot be used. Using the uncompact scan vectors solves the power problem, but then uses a lot of tester memory. Tester memory reloads are very time consuming and greatly increase test costs. Our goal was to study how scan vector compaction increases power and develop a procedure to compact scan vectors as much as possible without exceeding the constraints on average power and peak power. We wanted the procedure to be simple and fast so that it can be used on large industrial circuits without modifying them and easily incorporated in the conventional test generation flow used in industry.

Note that it is always possible to reduce average power during scan testing by simply scanning at a lower frequency, however, this comes at the cost of test time, and it does not help with the problem of peak power. Another solution is to add additional circuitry to the scan elements to hold the outputs at a constant value during scan. The drawback of this approach is the area and performance overhead that it incurs. The proposed static compaction procedure minimizes power without adding additional hardware.

2. Overview of Static Compaction

After an ATPG tool finds test cubes, *static compaction* is performed to reduce the number of test cubes. Static compaction involves trying to merge two compatible test cubes into one. Test cubes are compatible if in every bit position where either of the test cubes has a specified value (1 or 0), the other test cube either has the same specified value or an unspecified value (X). For example, 11XX0 is compatible with 1X0X0 but is not compatible with 011X1 because there is a conflict in the first and last bit position. A *conflict* arises when two test cubes have opposite specified values in a particular bit

position. Two compatible test cubes are merged by forming a single test cube in which only bit positions where both test cubes have X's remain as X's. In all other bit positions, one or both of the test cubes have a specified value, which is incorporated into the merged test cube. For example, merging 11XX0 with 1X0X0 forms the test cube 110X0. During static compaction, test cubes are considered two at a time, and compatible test cubes are merged to form a single test cube. This continues until no pair of test cubes can be merged.

After static compaction is performed, the remaining X's are filled with specified values and the test vectors are fault simulated in reverse order. Any test vector that doesn't detect any new faults is removed from the test set. The conventional approach for filling the X's in the test cubes is to do a *random fill (R-fill)*. R-fill involves replacing the X's randomly with 1's and 0's. The idea behind R-fill is that it increases the chance of detecting additional faults with a single test cube to hopefully eliminate the need for other test cubes so they will be dropped from the test set when it is reverse fault simulated. R-fill is very bad in terms of power, however. Randomly filling the X's may result in a lot of transitions when scanning the vectors into the scan chain, which may result in a lot of switching activity in the circuit. When power is a consideration, it is generally better to use a *minimum transition fill (MT-fill)*. MT-fill involves filling strings of X's with the same value to minimize the number of transitions. For example, when filling the test cube 01XX10, it would be best to fill the string of X's with 1's, i.e., 011110. For each string of X's in a test cube, if the specified bits on either side of the string have the same value, then the string of X's should be filled with that value to minimize the number of transitions. If they have opposite values, then it doesn't matter which value the string of X's is filled with. For example, when filling 0XX01X1X0, the first two X's should be filled with 0's, the third X should be filled with a 1, and the last X could be filled randomly with either 0 or 1. While MT-fill minimizes power, the drawback is that it may not be as effective as R-fill for detecting additional faults. Consequently, the reverse fault simulation step may not drop as many test vectors from the test set when MT-fill is used compared to when R-fill is used.

If the initial test cubes are not statically compacted, but simply filled with MT-fill, then the resulting test set will provide the minimum number of transitions per test vector for that initial set of test cubes. When two test cubes *a* and *b* are merged, the number of transitions in the resulting test cube *c* is never less than the maximum number of transitions in either *a* or *b* after MT-fill. This is because test cube *c* can be formed by specifying X's in either test cube *a* or *b*. There is no way to specify X's in

either test cube a or b so that there are fewer transition than MT-fill. Hence, the peak power is minimum for the initial test cubes, and it monotonically increases as static compaction is performed, i.e., it can never decrease.

Before presenting our static compaction procedure for minimizing power, we need to first discuss how we are estimating power.

3. Estimating the Power for Scan Vectors

In a CMOS circuit, there is very little static power dissipation. The predominant fraction of the power is dissipated when the circuit elements switch from logic 1 to 0 or vice versa. Under test, the circuit is entirely controlled by the test vectors that are applied to it. The elements in the circuit-under-test (CUT) will switch when the primary inputs change value or when the scan flip-flops change value. If the primary inputs of the CUT are directly controllable from the chip pins, then the tester can apply some vector to them during scan in. If the CUT is an embedded module or core, then both its flip-flops and primary inputs may be fed from a scan chain. In [Wang 97b], techniques for choosing the best vector to apply to the primary inputs of the CUT during scan-in when they are controllable from the chip pins were described. In this paper, if the primary inputs of the CUT are controllable from the chip pins, we assume they are held constant during scan-in. In this case, all switching activity during scan-in is due to the transitions in the scan chain.

Consider the case of a CUT with four scan flip-flops and a vector 1011 being scanned in. Let the scan flip-flops initially be 0000. After the first clock when first input has been scanned in, the scan flip-flops will be 1000. The state of the first flip-flop has changed from 0 to 1. This change in the flip-flop will cause other gates in the CUT to switch. The number of circuit elements that switch depends on the actual circuit. At the second clock, the flip-flops will be 1100. This change now results in a change in the second flip-flop but no change in the first flip-flop. At the third clock, the flip-flops will be 0110. The first, second and third scan flip-flops now switch.

This process continues until the complete test vector has been scanned in. The test vector is then applied to the CUT and the output response is captured back in the scan chain. As the next scan vector is scanned in, the output response from the previous vector is scanned out to the tester. Transitions in the output response being scanned out will also cause switching activity. We can divide the power dissipation during scan test into two parts. The scan-in power, which is due to transitions in the test vectors, and scan-out power which is due to transitions in the output response.

The proposed static compaction procedure attempts to find a test vector set that minimizes the power dissipated during scan testing. In order to do this, a means for comparing the power dissipated by two vectors is needed. The most accurate results would be obtained by using a circuit simulator to actually find the number of circuit elements that switch when a vector is scanned in. This process of using a simulator is expensive in terms of execution time. A simple heuristic is needed for comparing the power dissipated by two vectors. We present such a heuristic below.

Every vector does not dissipate the same amount of power. Consider the scan vector in Fig. 1. It has two transitions. When this vector is scanned into the CUT, Transition 1 passes through the entire scan chain. This transition dissipates power at every flip-flop in the scan chain. On the other hand, Transition 2 dissipates power only at the first scan flip-flop during scan in. The number of scan flip-flop transitions caused by a transition in a test vector being scanned in depends on its position in the test vector. In this example where there are 5 scan flip-flops, a transition in position 1 (which is where Transition 1 is) would be weighted 4 times more than a transition in position 4 (which is where Transition 2 is). The weight assigned to a transition is the difference between the size of the scan chain and the position in the vector in which the transition occurs. Similar reasoning can be applied for scan-out vectors.

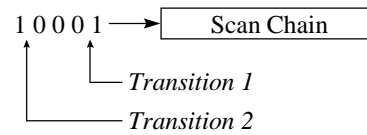


Figure 1. Transitions in Scan Vector

Hence, the power dissipated when applying two vectors can be compared by counting the number of weighted transitions in the vector. The number of weighted transitions is given by:

$$\text{Weighted_Transitions} = \sum (\text{Size_of_Scan_Chain} - \text{Position_of_Transition})$$

Given a vector, we apply the weighting rule and derive the number of weighted transitions. If vector a has more weighted transitions than vector b , then vector a is assumed to dissipate more power than vector b .

We present a graph to validate this heuristic. A series of test vectors was applied to a circuit using a digital circuit simulator that simulates the scan-in and scan-out operations. As the vectors were applied, we counted the number of circuit elements (gates) in the circuit that changed state. This is a measure of how much power would be actually dissipated in the circuit. We then

calculated the number of weighted scan-in transitions and weighted scan-out transitions. We plotted a graph of the sum of average weighted scan-in transitions and average weighted scan-out transitions versus average the number of circuit elements that make transitions in the CUT. A representative plot for one of the benchmark circuits is shown in Fig. 2. The two quantities are fairly closely correlated. In our algorithm, when we estimate the power dissipated by a vector, we use the number of weighted transitions as the measure. For the remainder of the paper, we will measure power in terms of flip-flop transitions.

The degree of freedom that we have in static compaction is in how to specify the X's in the test cubes. Trying to minimize both the scan-in and scan-out power during static compaction is an exceedingly difficult problem. Scan-in transitions can be directly controlled by filling the X's in the test cubes in a way that reduces the number of transitions. Scan-out vectors are dependent on the scan-in vectors and the relationship between the two depends on the function implemented by the CUT. Determining the effect of filling X's in the test cubes on the scan-out vectors requires circuit simulation. In order for the static compaction procedure to be efficient enough to use for industrial designs, we cannot afford the time required to compute the effect of each decision on the scan-out vectors. Hence, we just focus on minimizing scan-in power.

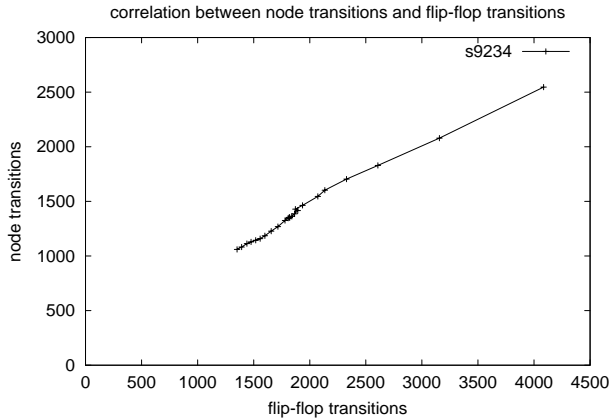


Figure 2. Correlation Between Node Transitions and Flip-Flop Transitions

4. Procedure for Minimizing Power

During static compaction, test cubes are merged and unspecified X values get specified. Let $Tran_Count(A)$ be the number of transitions in test cube A , and $Tran_Count(B)$ be the number of transitions in test cube B , then if test cube A and B are merged to form test cube C , then the number of transitions in test cube C , $Tran_Count(C)$, will be greater than or equal to the

maximum of $Tran_Count(A)$ and $Tran_Count(B)$:

$$Tran_Count(C) \geq \text{MAX} [Tran_Count(A), Tran_Count(B)]$$

In some cases, the number of transitions in the merged test cube can be much larger than in either of the original test cubes. Consider the case where the test cube 0X0X0X is merged with X1X1X1 to form 010101. The original test cubes had 0 transitions, but the merged test cube has 5 transitions.

Conventional static compaction procedures randomly merge compatible test cubes. However, merging some pairs of test cubes can result in a test cube with a large number of transitions whereas merging others would not increase the number of transitions by much. If the wrong test cubes are merged, the power can increase dramatically. So the idea behind our proposed static compaction procedure is to direct the process of selecting which test cubes to merge in a way that avoids generating merged test cubes with large numbers of transitions. In so doing, we try to minimize power in the final test set.

Our procedure for static compaction is as follows:

```
Create_Cost_Graph();
While vectors_can_be_combined()
  Select_a_pair_to_be_combined();
  Combine_the_vectors();
  Update_Cost_Graph();
```

We begin by forming a cost graph. There is one node for each test cube. For each pair of compatible test cubes (a, b) , an edge is placed between the corresponding nodes. The weight attached to the edge is the increase in power that results from removing a and b from the test set and applying the merged vector ab instead.

In each iteration of the procedure, we select a pair of test cubes to be combined. The objective is to pick pairs of test cubes so that the average power or peak power is minimized. The problem is complex as the graph changes each time a test cube pair is merged because two nodes are replaced by a single node. The procedure chooses the test cube pair using a greedy heuristic. The pair of nodes with the smallest edge weight is selected in each iteration. When two test cubes are combined, the graph can be quickly updated. The old nodes and their edges to other nodes are no longer valid. The two nodes are removed and a new node representing the combined vector is created. The procedure continues until no two vectors can be combined.

If there is a constraint on average power during test for some CUT, the average power can be monitored during each iteration of the static compaction procedure and the procedure can stop if the average power reaches a certain threshold.

If there is a constraint on peak power for some CUT, the procedure is modified in the following way. When

the edges are added to the cost graph, a check is made to see if merging a pair test cubes would cause the peak power to exceed the constraint. If so, then the edge between the corresponding nodes is not added to the graph. This prevents the static compaction procedure from violating the peak power constraint.

If there are N test cubes, then the cost graph has $N(N-1)/2$ nodes. Thus the procedure is quadratic in the number of test cubes. If the number of test cubes is large, the procedure can be sped up by not constructing the full graph. A graph that is linear in the number of test cubes can be constructed instead as follows. In every iteration, one test cube is randomly selected. Only edges between the selected test cube and all the other test cubes are computed. The minimum cost edge is then selected.

5. Experimental Results

We implemented the proposed static compaction procedure and compared it with conventional static compaction where test cubes are randomly merged. In both cases, all remaining X's were filled with Minimum Transition Fill as described in Sec. 2. The number of vectors remaining and the average weighted scan-in power and the peak power was recorded at every step in the algorithm. The results for some of the largest ISCAS 89 circuits are shown in the figures.

We show two plots for each of the circuits. The first is a plot of average power versus the number of test vectors. As the number of vectors decreases the average power increases. This is because the average number of transitions per vector increases due to the merging of vectors. For all circuits, the average power of the proposed static compaction procedure is much less than that of conventional static compaction. All the circuits have a characteristic "knee" in the graph where the average power rises rapidly with a small decrease in the number of test vectors. Looking at graphs like these, the designer can decide to sacrifice a small amount of compression and gain a lot in power savings. We have marked a point in each the graph where there is a large difference (2 to 3 times) in the power between conventional static compaction and the proposed procedure.

We show another plot for each circuit. It is the peak power of the test set versus number of vectors. The peak power is the maximum number of transitions that occur in a single test vector in the entire test set at that stage. The average number of transitions per vector increases when vectors are combined. Hence as expected, the peak power increases as the number of vectors decrease. For all circuits the proposed static compaction procedure has much lower peak power compared to conventional static compaction.

6. Conclusions

A simple yet effective approach was presented for generating scan vectors with significantly reduced average power and peak power. The static compaction procedure described here can be used to satisfy constraints on average power and peak power for scan testing. It can be easily implemented in the standard test generation flow used in industry today.

As integration density and clock frequencies continue to rise, power consumption during test may become a major constraint that limits test time. Simple techniques for reducing power such as the one presented here will be very useful for minimizing power during test and allowing more modules to be tested concurrently.

Acknowledgements

This material is based on work supported in part by the National Science Foundation under Grant No. MIP-9702236, and in part by the Texas Advanced Research Program under Grant No. 003658-0644-1999.

References

- [Chou 94] Chou, R.M., K.K.Saluja, and V.D. Agrawal, "Power Constraint Scheduling of Tests," *Proc. of International Conference on VLSI Design*, pp. 271-274, 1994.
- [Crouch 99] Crouch, A. *Design-For-Test for Digital IC's and Embedded Core Systems*, Prentice Hall, ISBN 0-13-084827-1.
- [Dabholkar 98] Dabholkar, V., S. Chakravarty, I. Pomeranz, and S.M. Reddy, "Techniques for Minimizing Power Dissipation in Scan and Combinational Circuits During Test Application," *IEEE Trans. on Computer-Aided Design*, Vol. 17, No. 12, pp. 1325-1333, Dec. 1998.
- [Devadas 95] Devadas, S., S. Malik, "A Survey of Optimization Techniques Targeting Low Power VLSI Circuits," *Proc. of Design Automation Conference*, pp. 242-247, 1995.
- [Gerstendörfer 99] Gerstendörfer, S., and H.-J. Wunderlich, "Minimized Power Consumption for Scan-Based BIST," *Proc. of International Test Conference*, pp. 77-84, 1999.
- [Girard 99] Girard, P., L. Guiller, C. Landrault, and S. Pravossoudovitch "A Test Vector Inhibiting Technique for Low Energy BIST Design," *Proc. of VLSI Test Symposium*, pp. 407-412, 1999.
- [Goel 81] Goel, P., "An Implicit Enumeration Algorithm to Generate Tests for Combinational Logic Circuits," *IEEE Trans. on Computers*, Vol. C-30, No. 3, pp. 215-222, Mar. 1981.
- [Hertwig 98] Hertwig, A., and H.-J. Wunderlich, "Low Power Serial Built-In Self-Test," *Proc. of European Test Workshop*, pp. 49-53, 1998.
- [Wang 94] Wang, S., and S.K. Gupta, "ATPG for Heat Dissipation Minimization during Test Application," *Proc. of International Test Conference*, pp. 250-258, 1994.
- [Wang 97a] Wang, S., and S.K. Gupta, "DS-LFSR: A New BIST TPG for Low Heat Dissipation," *Proc. of Int. Test Conf.*, pp. 848-857, 1997.
- [Wang 97b] Wang, S., and S.K. Gupta, "ATPG for Heat Dissipation Minimization for Scan Testing," *Proc. of Design Automation Conference*, pp. 614-619, 1997.
- [Wang 99] Wang, S., and S.K. Gupta, "LT-RTPG: A New Test-Per-Scan BIST TPG for Low Heat Dissipation," *Proc. of International Test Conference*, pp. 85-94, 1999.
- [Zorian 93] Zorian, Y., "A Distributed BIST Control Scheme for Complex VLSI Devices," *Proc. of VLSI Test Symposium*, pp. 4-9, 1993.

Figure 3.
Results for *s9234*

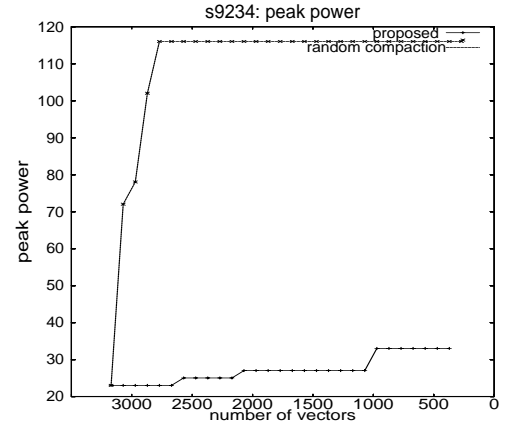
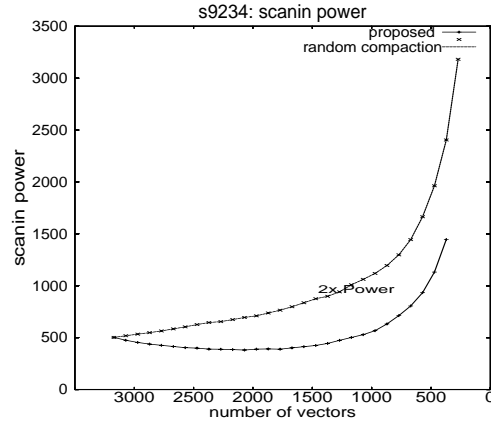


Figure 4.
Results for *s13207*

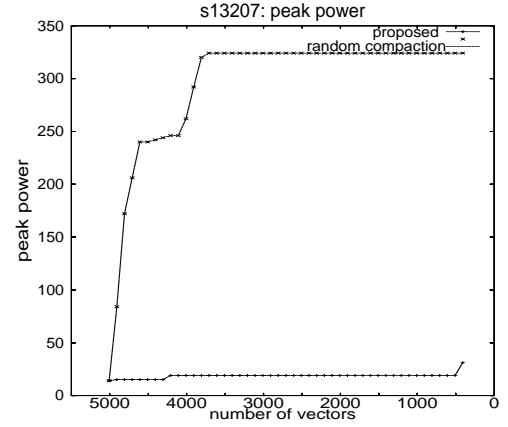
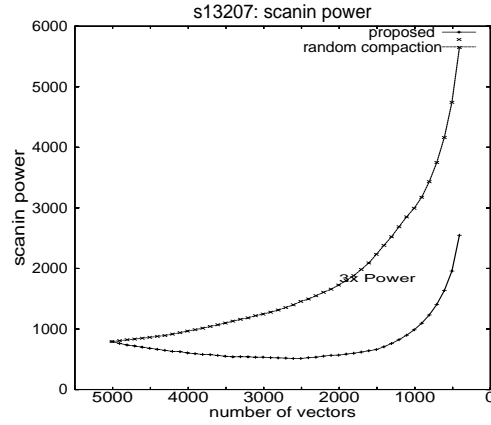


Figure 5.
Results for *s15850*

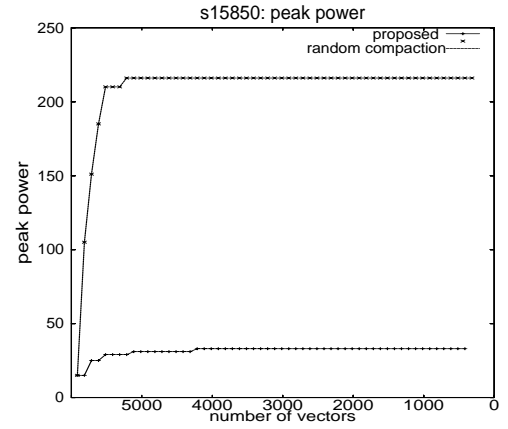
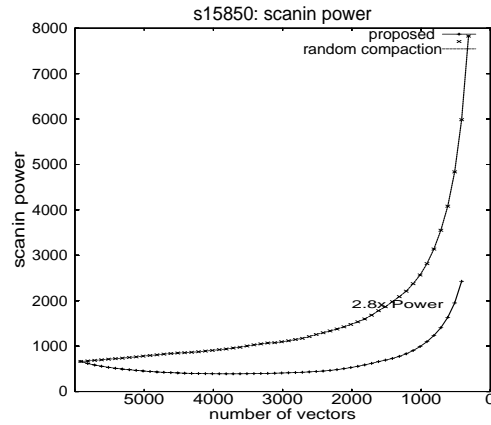


Figure 6.
Results for *s38584*

