Joint Minimization of Power and Area in Scan Testing by Scan Cell Reordering

Technical Report: UT-CERC-TR-NAT02-1

Shalini Ghosh[1], Sugato Basu[2] and Nur A. Touba[1]

[1]Computer Engineering Research Center
Dept. of Electrical and Computer Engineering
University of Texas, Austin, TX 78712-1084
E-mail: {shalini, touba}@ece.utexas.edu

[2]Dept. of Computer Sciences
University of Texas,
Austin, TX 78712-1084
Email: sugato@cs.utexas.edu

Abstract

This paper describes a technique for minimizing power dissipation that is also capable of reducing the area overhead of the circuit, compared to a random ordering of the scan cells. For a given set of test-vectors, we find the (locally) optimal re-ordering of the scan cells that minimizes a score function, which is a linear combination of the power and the area overhead. The score function has a trade-off parameter l *that can be used by the designer to specify the relative importance of area overhead minimization and power minimization. Our proposed greedy algorithm finds the best ordering for a given value of* l*. The strength of our algorithm lies in the fact that we use a novel dynamic minimum transition fill (MT-fill) technique for the unspecified bits in the test vector. Experiments performed on the ISCAS-89 benchmark suite show a reduction in power (e.g., 70% for circuit s13207,* l *= 500) as well as a reduction in layout area (e.g., 6.72% for circuit s13207,* l *= 500).*

1. Introduction

Two of the main drawbacks of scan testing [McCluskey 86] are the power dissipation and the area overhead. In CMOS circuits, power dissipation is proportional to the amount of switching activity that takes place [Devadas 95]. During scan testing, a much larger percentage of the scan cells will change value in each clock cycle than during normal operations. This excessive switching activity during scan testing can cause power dissipation in the circuit to be very high. Another drawback of scan testing is the area overhead. As suggested in [Makar 98], one of the biggest components adding to the area overhead is the stitching wire between consecutive cells in the scan chains. Different techniques for controlling power dissipation during testing have been proposed in [Dabholkar 98], [Sankaralingam 00], [Girard 99], [Gestendorfer 99], [Chow 94], [Wang 97A], [Wang 97B], [Hertwig 98], [Wang 99], while various techniques for reducing the area overhead have been proposed in [Makar 98], [Lin 96].

In this paper, we describe a technique for minimizing power dissipation during scan testing. Our technique is also capable of reducing the area overhead of the circuit, with respect to random ordering of the scan cells. For a given set of test-vectors, we find the (locally) optimal re-ordering of the scan cells that minimizes a score function, where the score function is a linear combination of the power and the area overhead. The score function has a trade-off parameter λ that can be used by the designer to specify the relative importance of area overhead minimization and power minimization – increasing λ causes a decrease in the power dissipation in the circuit, at the cost of increased area overhead. We propose a greedy algorithm for finding the best ordering for a given value of λ. The strength of our algorithm lies in the fact that we use a novel dynamic minimum transition fill (MT-fill) of the 'X' (i.e. unspecified) bits in the test vector. The method of doing "on-the-fly" MT-fill of the test vector matrix while calculating the optimal ordering gives us a better power reduction in the re-ordered matrix, details of which will be explained in Section 4.

We ran experiments on standard benchmark circuits and show power versus area overhead trade-off plots. These plots provide the designer with the flexibility of giving more importance to minimizing power or area overhead, according to the design requirements, by choosing a suitable value of the design parameter λ.

Our experiments show that with a proper choice of the parameter λ, our algorithm is quite effective in reducing the power in a circuit. For example, power in the circuit s13207 was reduced by 70%, for λ = 500. It is also capable of reducing the area overhead of the circuit, with respect to random ordering of the cells. For example, layout area

overhead in the circuit s13207 was reduced by 6.72%, for $\lambda = 500$.

The rest of the paper is organized as follows. Section 2 gives a background on concepts used in this paper, Section 3 explains our scan re-ordering methodology for minimizing power and area overhead, Section 4 explains our proposed algorithm for choosing the best ordering of the scan cells, Section 5 gives the experimental methodology, and Section 6 discusses the experimental results.

2. Background

In this section, we discuss some of the relevant background concepts used in the paper.

2.1 Estimation of Power

In CMOS circuits, the predominant fraction of power is dissipated when circuit elements switch from logic 0 to 1 or vice versa. For a circuit-under-test (CUT), controlled entirely by the test vectors applied to it, the elements will switch value when the primary inputs change value or the scan cells change values. In this paper we assume that if the primary inputs of the CUT are directly controllable from the chip pins, then they are held constant during scan-in. Thus in this case, during scan-in, all switching activity is due to the transitions in the scan chain.

Let us consider a CUT with 4 scan cells and a vector 1001 being scanned in. Let the scan cells initially be 0000. At the first clock when the first input is scanned in, the state of the scan cells will be 1000. Thus the state of the first cell has changed from 0 to 1.This will cause other gates in the CUT to switch depending on the circuit. At the second clock when the next input is being scanned in, the cells will be in a state 0100. Here both the first and the second cells have changed states. This 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 being scanned in, the transitions in the output response from the previous scan vector being scanned out will also cause switching activity. Thus we can divide the power dissipation during scan testing into:

- scan-in power - due to transitions in scan test vectors
- scan-out power - due to transitions in the output response being scanned out

The best way to estimate power during scan testing would be to do actual circuit simulation to actually find the number of circuit elements that switch when a vector is scanned in. However this procedure takes a very long execution time and is thus very expensive. Instead, in this paper to estimate the scan-in or the scan out power, we use the weighted transitions metric proposed by [Sankaralingam 00]. In their paper, they have found that the sum of average weighted scan-in transitions and the average weighted scan-out transitions is fairly closely correlated to the average number of circuit elements that make transitions in the CUT. The weighted transitions metric model can be explained as follows.

Consider the previous example of the scan-in vector 1001. As shown in Fig.1, there are two transitions in the scan vector. While Transition 1 dissipates power at every cell in the scan chain while being scanned in, Transition 2 only dissipates power at the first scan cell. Thus when a test vector is being scanned in, the number of scan cell transitions caused by a particular transition in that vector would depend on the position of the transition in the scan vector. According to [Sankaralingam 00], 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. The number of weighted transitions is given by:

$$\text{Weighted\_Transitions} = \Sigma\,(\text{Size\_of\_Scan\_Chain} - \text{Position\_of\_Transition})$$

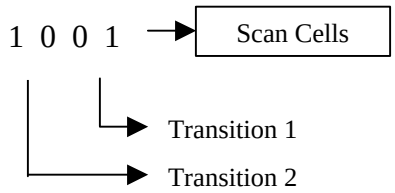

Figure 1. Transitions in example scan vector [Sankaralingam 00]

2.2 Estimation of Area overhead

In an algorithm that re-orders the scan chains to reduce the power dissipation, the main concern is whether the re-ordering increases the area overhead of the circuit. In this paper we have used two terms that give a measure of the area overhead. One measure of the area overhead is the layout area, which measures the overall area of the chip. Another heuristic measure of area that we use in this paper is the approximate area overhead, which is the sum of the Manhattan distance between two consecutive cells in the scan chain. This is an estimate of the stitching-wire length between consecutive cells in the scan chain, and is one of the biggest components to the area overhead [Makar 98]. Manhattan distance gives the estimate of the routing complexity – the longer the Manhattan distance, the greater is the routing complexity and the designer cannot compact area as much. So we have chosen the Manhattan distance as an estimate for approximate area overhead.

2.3 Minimum-transition Fill (MT-fill)

One important feature of our optimization algorithm is the fact that we do MT-fill of test vectors "on-the-fly", which we call dynamic MT-fill. Details of this method will be explained in Section 4. In this section, we give a background of MT-fill.

Consider a test vector matrix that has 0, 1 and X entries, where each row of the matrix corresponds to a test vector for the circuit. X is an unspecified value and can be filled with either 0 or 1. The conventional approach for filling the X's in the test cube is to do random fill (R-fill) in which the X's are randomly replaced by 0's or 1's. In R-fill, the idea is that it increases the chance that a single test cube would detect additional faults and hopefully the other test cubes would not be required and can be eliminated during reverse fault-simulation. However, since we are considering power, which involves the number of weighted transitions in the test vector, it is best to consider Minimum Transition Fill (MT-fill). In MT-fill, a series of X entries in the test vector are filled with the same value as the first non-X entry on the right side of this series. This minimizes the number of transitions in the test vector when it is scanned in. For example, consider the test vector: 100**XX**010**X**1**X**0. This vector, after MT-fill, would become 100**00**010**1**1**0**0. If the test vector has a string of X bits that is not terminated by a non-X bit on the right side, then it should be filled by the bit value to the left of the sequence. For example: 1000001011**XX** should be 100000101**11** after MT-fill.

3. Scan Reorder Methodology for Minimizing Power and Area overhead

In the conventional approach, a gate level netlist is generated after design synthesis. After that, scan insertion is done and then placement and routing is done. In our scan reorder methodology, the conventional process is modified as shown in Fig. 2.

In our methodology, after scan insertion we do placement to get the initial co-ordinates of the scan cells in the circuit. These co-ordinates are used for finding the best ordering of the scan cells using our proposed algorithm, which tries to do joint minimization of power and area overhead. The scan chain cells are re-connected according to this new ordering, after which placement and routing is done again to get the new co-ordinates of the re-ordered scan cells. The test vectors are also reordered according to the new ordering of the scan cells.

4. Algorithm for Ordering Scan Cells

As mentioned earlier, in our algorithm (shown in Fig. 3), we start with a test vector matrix that has 0, 1 and X entries, where each row of the matrix corresponds to a test vector for the circuit. Each test vector column corresponds to a scan cell in the scan chain, such that ordering the columns in the test vector matrix is equivalent to ordering the scan cells in the scan chain.

In order to find the best ordering of the scan cells in the scan chain, we try to jointly minimize power and area overhead and study a trade-off between the two. For that, we define a score function between two columns *i* and *j* of a test vector matrix:

$$Score(i,j) = Distance(i,j) + l * Power(i,j)$$

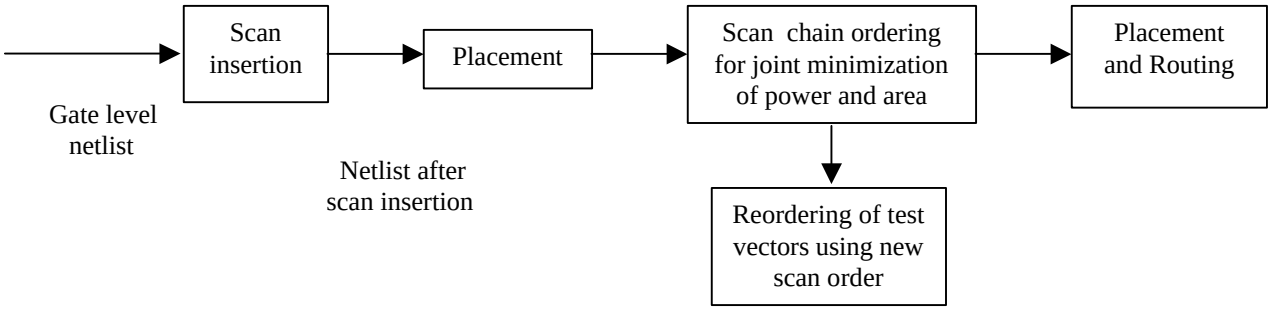

Figure 2. Scan Cell Reordering Methodology

where λ is the trade-off parameter between distance and power. *Distance(i,j)* is measured by the Manhattan distance between the scan cells corresponding to columns *i* and *j* (as explained in Section 2.2), while *Power(i,j)* is measured by weighted transitions (as explained in Section 2.1).

For a given value of λ, the algorithm starts by choosing the 2 columns in the test matrix that have the minimum score between them, in the function *findBestSeeds*. These two columns are placed on the rightmost part of the test matrix, as "seeds". This is because, considering that the test vectors are inserted into the scan chain from the left, the transitions between the two columns on the rightmost side of the test matrix would have the maximum weight (as explained in Section 2.1). So, we want to greedily seed the re-ordering process by assigning the two columns with minimum score between them as the rightmost columns in the re-ordered matrix. After that, the column before the rightmost one is MT-filled with respect to the rightmost column. We refer to this process of doing MT-fill "on-the-fly" as *dynamic MT-fill.*

We illustrate this technique of dynamic MT-fill with the following example. After selecting the best two columns for seeding, the algorithm places them as the rightmost two columns of the test matrix. After this step let the example test vector matrix look like Fig 4 (a). The algorithm does dynamic MT-fill at this stage. The current column under consideration (the second column from the right in this example) gets changed by specifying the X's such that the number of transitions between the current column and the rightmost column is minimized. The resulting matrix is shown in Fig 4 (b).

In the main loop of the function *findBestOrdering*, we consider every column *i* in the matrix from right to left (due to the weighting scheme), starting from the column before the rightmost one. The function *greedyColumnToSwap* in the algorithm finds the column *j* in the matrix, to the left of *i*, which has the minimum score with column *i*. It then swaps column *i-1* with column *j*. After swapping, the new column *i-1* is dynamically MT-filled with respect to column *i*.

The dynamic MT-fill done at every step of the re-ordering process is a novel technique, which greatly contributes to the improved performance of our algorithm. Instead of doing the dynamic MT-fill at every step, if we had performed the MT-fill on the total test vector matrix at the beginning of the algorithm, then we would have lost degrees of freedom in choosing the best values with which to fill up the X values in the test vectors in order to minimize the number of transitions. If, on the other hand, we had chosen to do a MT-fill at the end of the algorithm, then we would have too many options for selecting the best column at every step of the algorithm. Dynamic MT-fill gives a good compromise between these two extremes, by

```
Input:- TestMatrix[rowSize,columnSize]
        // Test vector matrix
        - LocationsVector[columnSize]
        // Contains (X,Y) co-ordinates of the scan cells
        - LambdaVector[numLambdas]
        // Contains diff. values of parameter lambda

Output: Best column ordering for each lambda value

findBestOrdering() {
  for each lambda in LambdaVector {
    findBestSeed(lambda);
    Dynamic MT-fill column (columnSize-1) w.r.t.
      column columnSize;
    for (i=columnSize-1; i>=2; i--) {
      bestColumn =
        greedyBestColumnToSwap(i,lambda);
      Swap column bestColumn with column (i-1);
      Dynamic MT-fill column (i-1) w.r.t. column i;
    }
    Output the column ordering
  }
}

findBestSeed(lambda) {
  Find the pair of columns [a,b] from TestMatrix
    that have minimum score(a,b,lambda)
  Swap column pair (a,b) with the two rightmost
    columns of TestMatrix
}

greedyBestColumnToSwap(i,lambda) {
  Find the column j in the TestMatrix that has
    the minimum score(i,j,lambda)
}

score(i,j,lambda) {
  Between columns i and j in TestMatrix, compute
    [wireLength(i,j)+lambda*transitions(i,j)]
}

wireLength(i,j) {
  Compute Manhattan distance between cell i
    and cell j, using LocationsVector
}

transitions(i,j) {
  Compute weighted 0->1 or 1->0 transitions
    between column i and column j of the TestMatrix
}
```

Figure 3. Algorithm for Scan Cell Reordering

selecting the X values in the current column so as to minimize the number of transitions at every step of the algorithm, while at the same time keeping enough degrees of freedom for selecting the best columns in the future steps.

At every step of the algorithm, the overall score (power and distance) between the test matrix columns is greedily minimized. At the end of the swaps, the new column ordering is output. This process is repeated for all values of the trade-off parameter λ that are specified by the designer.

1 1 2 0	1 1 **0** 0
2 1 2 0	2 1 **0** 0
0 1 0 0	0 1 **0** 0
1 0 2 1	1 0 **1** 1
(a)	(b)

Figure 4. Dynamic MT-fill of column 3 w.r.t. column 4

5. Experimental Results

The experimental methodology is as follows:

1. For a given circuit, insert the scan chain into it.

2. Run a placement and routing tool to get the locations of the scan cells and the total area overhead of the circuit.

3. Using the test-vectors and the locations of the cells, run the greedy algorithm to get the (local) optimal ordering of the cells for different values of the scaling parameter λ.

4. From Step 3, we get the approximate area overhead and estimated power of the circuit corresponding to the ordering for a particular value of λ. These values are used to plot the power and the approximate area overhead for each value of λ.

5. After looking at the plots in Step 4, we choose a particular value of λ that gives a good trade-off between power and area overhead. For this value of λ, we stitch the scan-chain according to the ordering found in Step 3. Running an area-measurement tool gives the layout area overhead for this λ.

We performed experiments on the following circuits from the ISCAS-89 benchmark suite [Brglez 89]: *s5378*, *s9234*, *s13207*, *s15850* and *s38417*. We used the *wolfe* tool [Sechen 85] in OctTools for routing, placement, and calculating the area overhead of the circuit after placement.

The results on the five benchmark circuits are shown in Figs. 4-8. In each figure, the initial estimated area of the circuit refers to the approximate area overhead of the circuit with the default ordering of the scan cells. The approximate area overhead is calculated as the sum of the lengths of the wires connecting the scan cells, i.e., the Manhattan distance between scan cells (as explained in Sec. 2.2). The initial estimated power is the power estimated from the number of transitions in the test vectors with the default ordering. For each value of λ, our algorithm finds the (local) optimal ordering by minimizing the combined power and area overhead metric [Section 4]. For each λ, the final estimated area is the approximate area overhead of the circuit after layout and placement, with the scan cells being ordered according to the optimal ordering, while the final estimated power is the power estimated from the number of transitions in the test-vector matrix after the ordering of the scan cells.

As can be seen from the graphs of all the circuits, increasing the value of λ increases the final approximate area overhead and decreases the final estimated power, after the scan chain has been re-ordered using the ordering output by the algorithm. For all of our circuits, the designer can choose the value of λ to trade-off the savings in power with increase in approximate area overhead by looking at

Table 1. Results showing the reduction in estimated power and layout area overhead for the optimal chosen value of λ for the experimental benchmark circuits

Benchmark circuit	Size of scan chain	Chosen λ value	Reduction in estimated power	Reduction in actual layout area
s5378	164	100	48.89%	4.83%
s9234	211	100	47.17%	3.79%
s13207	638	500	70.20%	6.72%
s15850	534	500	58.29%	5.42%
s38417	1636	1000	61.50%	5.01%

the graphs. For each circuit, we chose a particular trade-off value of λ. For that value of λ, the percentage reduction of the actual layout area overhead of the circuit after scan-chain re-ordering was calculated, by running a placement and routing tool on the circuit with the reordered scan cells.

As can be seen from the graphs, our algorithm achieves very good reduction in estimated power and approximate area overhead for all the benchmark circuits.

For the trade-off value of λ chosen by us corresponding to each circuit, the percentage reduction in estimated power and actual layout area is shown in Table 1.

6. Conclusion

We have considered a technique of minimizing power dissipation in a circuit by re-ordering scan cells. Our method also reduces the area overhead in the circuit, compared to random ordering. For a given set of test-vectors, our greedy algorithm finds the (locally) optimal re-ordering of the scan cells which minimizes a score function, where the score function is a linear combination of the power and the approximate area overhead – increasing λ causes a decrease in the power dissipation in the circuit, at the cost of increased approximate area overhead. The strength of our algorithm lies in the fact that we use a novel dynamic Minimum Transition fill technique for the unspecified bits in the test vector, which gives us a better power reduction in the reordered test-vector matrix.

Our experiments on circuits from the ISCAS-89 benchmark suite show that with a proper choice of the parameter λ, our algorithm is quite effective in reducing the power in a circuit. For example, power in the circuit s13207 was reduced by 70%, for λ = 500. It is also capable of reducing the area overhead of the circuit, with respect to random ordering of the cells. For example, area overhead in the circuit s13207 was reduced by 6.72%, for λ = 500.

We also show how the power versus area overhead trade-off plots can provide the designer with the important flexibility of giving more importance to either minimizing power or minimizing area overhead, according to the design requirements, by choosing a suitable value of the design parameter λ.

References

[Brglez 89] F. Brglez, D. Bryan, and K. Kozminski, "Combinational profiles of sequential benchmark circuits", Proceedings of *International Symposium on Circuits and Systems*, pp. 1929-1934, 1989.

[Chow 94] R.M. Chow, K.K. Saluja and V.D. Agrawal, "Power Constraint Scheduling of Tests", *Proceedings of International Test Conference on VLSI Design*, pp. 271-274, 1994.

[Dabholkar 98] V. Dabholkar, S. Chakravarty, I. Pomeranz and S.M. Reddy, "Techniques for Minimizing Power Dissipation in Scan and Combinational Circuits During Test Application", *IE7EE Trans. on Computer-Aided Design*, Vol.17, No. 12, pp. 1325-1333, December, 1998.

[Devadas 95] S. Devadas and S. Malik, "A Survey of Optimization Techniques Targeting Low Power VLSI Circuits", *Proceedings of Design Automation Conference*, pp. 242-247, 1995.

[Gestendorfer 99] S. Gestendorfer and H.J. Wunderlich, "Minimized Power Consumption for Scan-Based BIST", *Proceedings of International Test Conference*, pp. 77-84, 1999.

[Girard 99] P. Girard, L. Guiller, C. Landrault and S. Pravossoudovitch, "A Test Vector Inhibiting Technique for Low Energy BIST Design", *Proceedings of VLSI Test Symposium*, pp. 407-412, 1999.

[Hertwig 98] A.Hertwig and H.J. Wunderlich, "Low Power Serial Built-In Self Test", *Proceedings of European Test Workshop*, pp. 49-53, 1998.

[Lin 96] K.H.Lin, C.S.Chen and T.T.Hwang, "Layout-driven Chaining of Scan Flip-Flops", *IEE Proceedings-Comput.Digit.Tech*., Vol. 143, No. 6, November 1996.

[Makar 98] S. Makar, "A Layout-Based Approach for Ordering Scan Chain Flip-Flops", *IEEE International Test Conference*, pp. 341-347, 1998.

[McCluskey 86] E.J. McCluskey, "Logic Design Principles", Prentice-Hall, NJ, 1986.

[Sankaralingam 00] R. Sankaralingam, R. Oruganti and N. Touba, "Static Compaction Techniques to Control Scan Vector Power Dissipation", *IEEE VLSI Test Symposium*, pp. 35-42, 2000.

[Sechen 85] C. Sechen and A. Sangiovanni-Vincentelli. "The TimberWolf Placement and Routing Package", *IEEE Journal of Solid State Circuits*, vol. 20, pp. 510-522, 1985.

[Wang 97A] S. Wang and S.K. Gupta, "DS-LFSR: A New BIST TPG for Low Heat Dissipation", *Proceedings of Int. Test Conference*, pp. 848-857, 1997.

[Wang 97B] S. Wang and S.K.Gupta, "ATPG for Heat Dissipation Minimization for Scan Testing", *Proceedings of Design Automation Conf*., pp. 614-619, 1997.

[Wang 99] S.Wang and S.K.Gupta, "LT-RTPG: A New Test-Per-Scan BIST TPG for Low Heat Dissipation", *Proceedings of International Test Conference*, pp. 85-94, 1999.

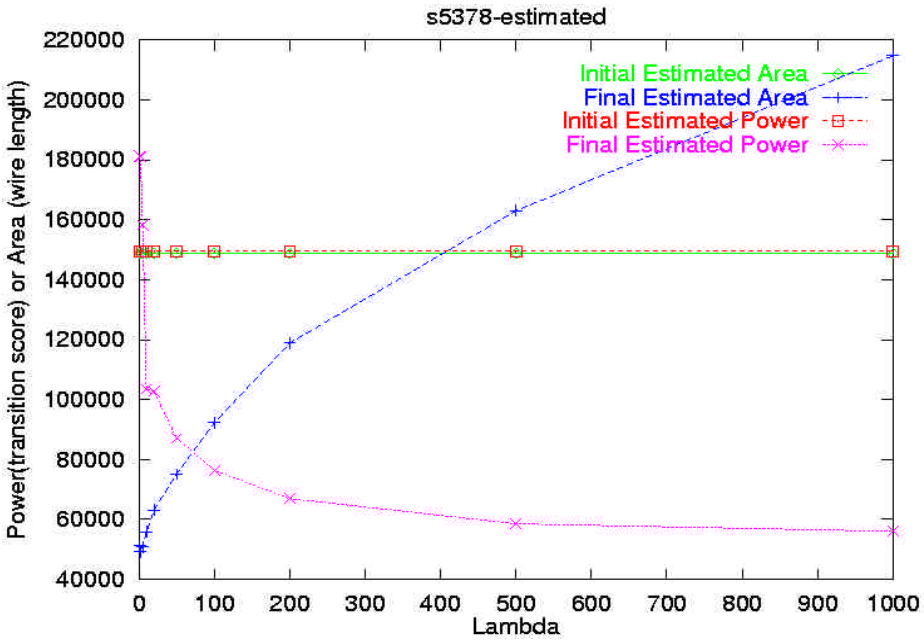

Figure 5. Results for s5378

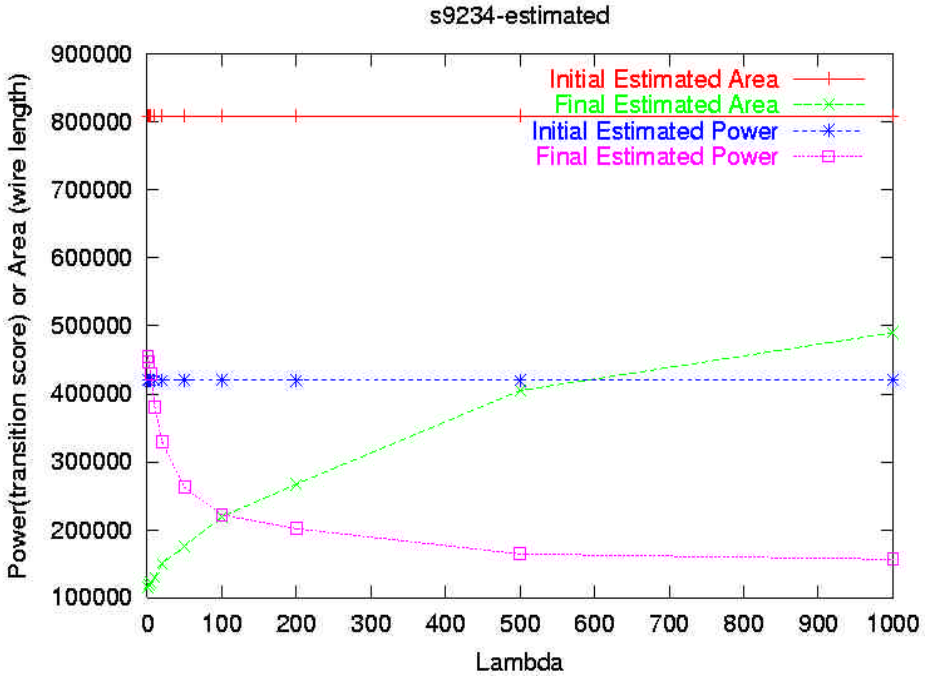

Figure 6. Results for s9234

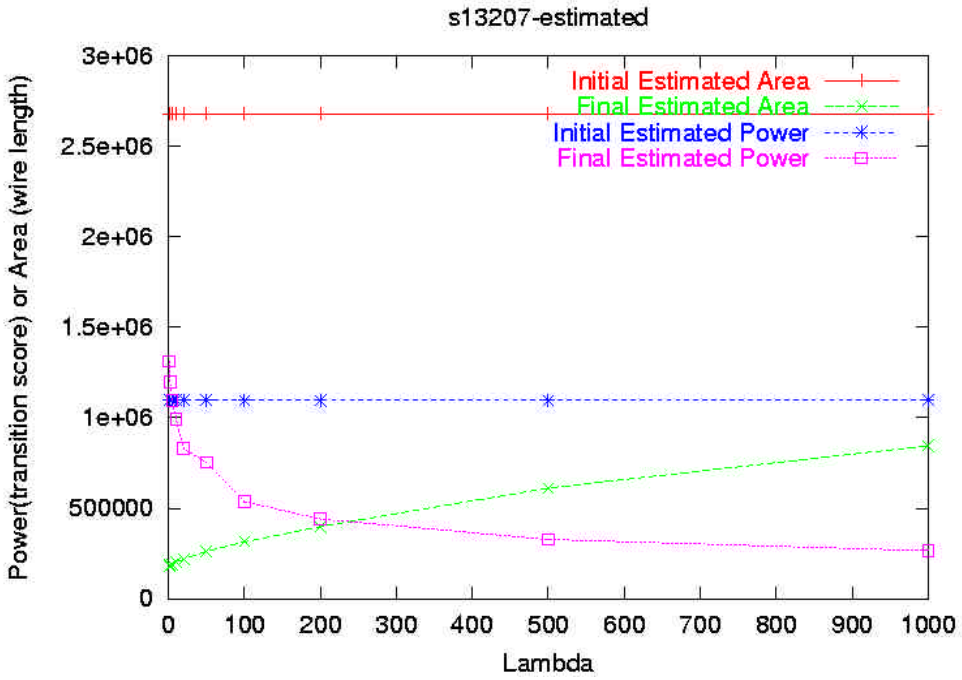

Figure 7. Results for s13207

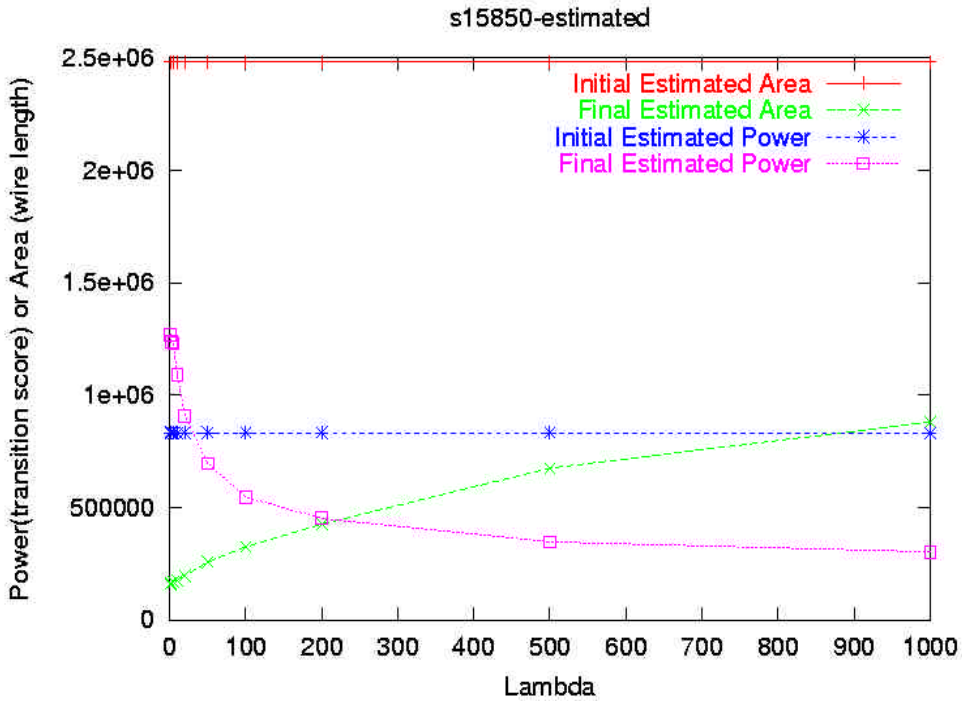

Figure 8. Results for s15850

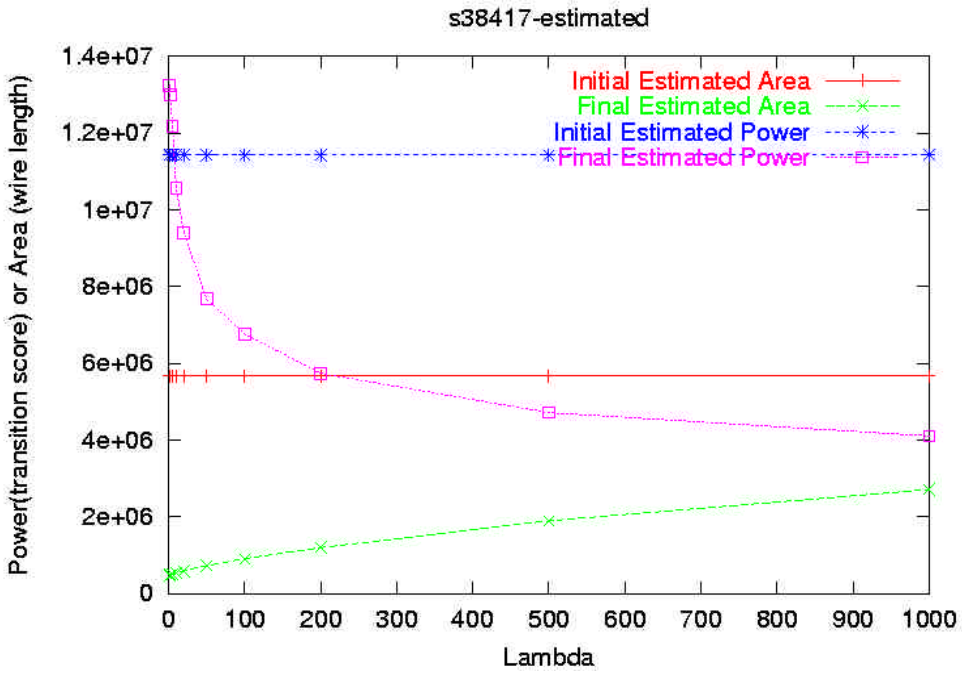

Figure 9. Results for s38417