Controlling Peak Power During Scan Testing

Ranganathan Sankaralingam and Nur A. Touba

Computer Engineering Research Center
Department of Electrical and Computer Engineering
University of Texas, Austin, TX 78712

Abstract

This paper presents a procedure for modifying a given set of scan vectors so that the peak power during scan testing is kept below a specified limit without reducing fault coverage. The proposed approach works for any conventional full-scan design -- no extra design-for-test (DFT) logic is required. If the peak power in a clock cycle during scan testing exceeds a specified limit (which depends on the amount of peak power that can be safely handled without causing a failure that would not occur during normal functional operation) then a "peak power violation" occurs. Given a set of scan vectors, simulation is done to identify and classify the scan vectors that are causing peak power violations during scan testing. The problem scan vectors are then modified in a way that eliminates the peak power violations while preserving the fault coverage. Experimental results indicate the proposed procedure is very effective in controlling peak power.

1. Introduction

The peak power drawn in a single clock cycle during scan testing can be much larger than during normal 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 resulting in excessive switching activity in the circuit. If a large number of flip-flops switch simultaneously in a clock cycle, this results in a large current spike. A chip's power supply/ground pins and distribution system may be designed for handling the peak power during normal operation, and it may not be able to handle the large peak power that could occur during scan testing. 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-lock loop (PLL) to malfunction). With the rapid increase in chip complexity, the problem of excessive peak power during scan testing is becoming an important issue in industry.

The average power dissipation during scan testing can be controlled by reducing the scan clock frequency, however, the peak power during scan testing is independent of clock frequency and hence is much more difficult to control. Moreover, controlling peak power requires ensuring that the peak power dissipation in any single clock cycle does not exceed the capabilities of the chip, which is more difficult than simply reducing the average power dissipation per clock cycle over the entire test session.

Previous work in low power scan testing has mostly focused on the problem of controlling heat dissipation by reducing the average power dissipation [Chou 94], [Wang 94, 97ab, 99], [Dabholkar 98], [Girard 99a], [Sankaralingam 00, 01], [Chandra 01]. Some design-for-test (DFT) techniques reduce peak power in addition to average power. In [Hertwig 98] and [Gerstendörfer 99], logic is added to hold the output of the scan cells at a constant value during scan shifting thereby reducing power dissipation. This approach greatly reduces average power, and will avoid peak power problems during scan shifting, but will not help with peak power problems during the capture cycle (which can arise if the circuit is placed in a state that it could not be in during functional operation). A drawback of this approach is that it degrades circuit performance because it adds extra logic in the functional paths. In [Whetsel 00], an adapted scan chain architecture that segments a single scan chain to minimize switching activity during scan shifting is proposed. This approach also greatly reduces average power and can avoid peak power problems during scan shifting, but it will not avoid peak power problems during the capture cycle and requires additional DFT hardware. In [Lee 00], a method for adding delay elements to interleave the capture cycles for multiple scan chains is proposed for reducing peak power during scan capture. In [Girard 99b], a technique for reducing peak power during built-in self-test (BIST) is proposed which involves partitioning a circuit and performing separate BIST sessions on each partition. In [Corno 99], a technique for modifying test sequences for sequential non-scan testing is proposed for reducing peak power.

In this paper, a procedure is proposed for modifying a given set of scan vectors so that the peak power during scan testing is kept below a specified limit

Proceedings of the 20 th IEEE VLSI Test Symposium (VTS'02)
1093-0167/02 $17.00 © 2002 IEEE

while maintaining the same fault coverage. No extra DFT hardware is required. The proposed approach works for any conventional full-scan design. The basic idea is that given a set of scan vectors that results in peak power violations during scan testing, simulation can be done to identify which scan vectors are causing the problem. The proposed procedure then modifies the problem scan vectors by reassigning input values in a way that reduces the amount of switching activity during scan testing while preserving the fault coverage. By so doing, peak power violations during scan testing can be eliminated.

2. Identifying and Categorizing Peak Power Violations

The peak power in each clock cycle during scan testing can be estimated through cycle by cycle simulation of the switching activity that occurs in the circuit. If the peak power in a clock cycle exceeds a specified limit (which depends on the amount of peak power that can be safely handled without causing a failure that would not occur during normal functional operation) then a "peak power violation" occurs in that clock cycle. During scan testing, each scan vector is scanned into the scan chain(s) and then there is a capture cycle in which the output response of the circuit is captured in the scan chain(s). The output response is then scanned out as the next scan vector is scanned in. There can be a peak power violation during either scan shifting or during scan capture. A peak power violation can occur during scan capture if a scan vector places the circuit in a state that it would never go into during normal functional operation, and that state caused peak power in excess of what could occur during normal functional operation.

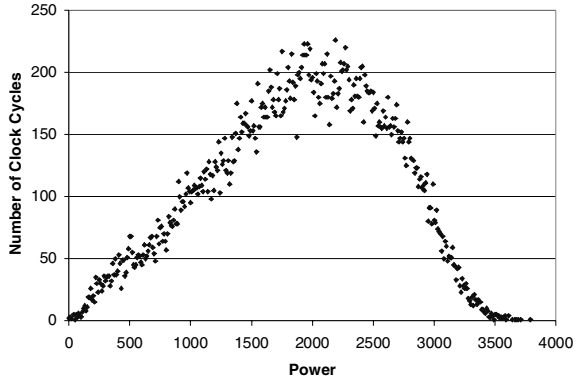

Figure 1. Distribution of Peak Power during Scan Testing of *s9234* Benchmark Circuit

Figure 1 shows a plot of the peak power as measured by the number of weighted gate transitions during each clock cycle when scan testing the *s9234* benchmark circuit. During each clock cycle, the number of gate transitions that occur was weighted by the number of fanouts of each gate. The plot in Fig. 1 shows the number of weighted gate transitions in the x-axis, and the number of clock cycles with that number of weighted gate transitions in the y-axis. Note that it looks like a normal curve which is a common characteristic for most circuits. The maximum peak power during scan testing occurs at the tail end of the curve and hence is caused by a relatively small number of clock cycles. Although the fraction of problem clock cycles is very small, these clock cycles may belong to a large fraction of the vectors in the total test set. Hence if the simple approach of dropping problem vectors is resorted to, a large decrease in fault coverage may result.

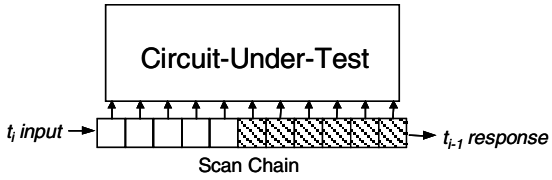

Figure 2. Scan vector t_i being scanned in and response of scan vector t_{i-1} being scanned out

Note that during scan shifting the scan chain contains part of the scan vector, t_i, being scanned in and part of the output response of the previous scan vector, t_{i-1}, which is being scanned out (as illustrated in Fig. 2). Consequently, the number of flip-flop transitions (and hence the amount of switching activity in the circuit as a whole) depends on the ordering of the scan vectors. For full scan, the ordering of the scan vectors does not affect the fault coverage, and hence changing the ordering of the scan vectors is one approach that can be used to reduce peak power (this will be explored in Sec. 3). Another approach to reduce peak power would be to place a "dummy vector" that has few or no transitions (e.g., a scan vector of all 0's) before or after some scan vector t_i so as to reduce the peak power that occurs when scanning in t_i or scanning out its output response. However, even with using dummy vectors before or after some scan vector t_i there still may be a peak power violation during scan shifting. Note also that adding dummy vectors may result in an unacceptable increase in test set size.

During simulation of scan testing, if there is a clock cycle in which a peak power violation occurs, we will classify the peak power violation in the following four ways:

1. *Scan capture problem*
2. *Order dependent problem* – scan shifting

problem which can be solved by using dummy vectors or reordering the scan vectors

3. *Scan-in problem* – scan shifting problem while scanning in the vector which cannot be solved by using a dummy vector
4. *Scan-out problem* – scan shifting problem while scanning out the output response which cannot be solved by dummy vectors

If the peak power violation occurs during scan capture, then it is a scan capture problem. If the peak power violation occurs during scan shifting, then further simulation is done using dummy vectors to classify it. Consider the case where the peak power violation occurs during scan shifting when the scan chain contains a part of scan vector, t_i, which is being scanned in, and a part of the output response of the previous scan vector, t_{i-1}, which is being scanned out (as illustrated in Fig. 2). We first simulate the case where scan vector t_i is shifted into the scan chain and the output response of the previous scan vector, t_{i-1}, is a dummy vector that causes few transitions as it is scanned out. If there is a peak power violation, then we classify scan vector t_i as a "scan-in problem." Then we simulate the case where the output response of scan vector, t_{i-1}, is scanned out, and a dummy vector that does not cause any transitions is scanned in (e.g., a vector of all 0's). If there is a peak power violation, then we classify scan vector t_{i-1} as a "scan-out problem." If there is neither a scan-in nor a scan-out problem, then we classify the scan vector pair (t_i, t_{i-1}) as an "order dependent problem" because the peak power violation could be eliminated by either reordering the scan vectors or placing a dummy vector between test vectors t_i and t_{i-1}.

So the overall procedure is to simulate the entire scan test and identify all the clock cycles in which peak power violations occur. We then classify each peak power violation as described above. The end result is a list of individual scan vectors that cause scan capture problems, scan-in problems, and scan-out problems, and a list of scan vector pairs that cause order dependent problems. The next section describes the procedure for how we modify the set of scan vectors to eliminate these peak power violations.

3. Overview of Procedure for Eliminating Peak Power Violations

After the peak power violations have been classified, the next step is to eliminate the problems. The order dependent problems are the easiest to solve since they do not necessarily require modifying the scan vectors themselves. However, the other problems require modifying the scan vectors. This needs to be done in a way that preserves the fault coverage. *Bit-stripping* is used to introduce unspecified values (i.e., X's) and then the values of the X's are reassigned in a way that reduces the peak power.

Bit-stripping is an operation performed on a scan vector, t. Fault simulation is done to drop all the faults that are detected by all the scan vectors in the test set except for t. Then t is fault simulated to determine the set of faults F_t that are only detected by t and by no other vector in the test set. The first bit in t is changed to an X and 3-valued fault simulation is performed to see if all the faults in F_t are still detected. If so, then the bit is kept as an X, otherwise it is returned to its previous value. This process is repeated for all the bits in t. The end result of the bit-stripping operation is that a number of X's are introduced into t without reducing the overall fault coverage of the test set. If the number of X's is still not sufficient, it can be increased by partitioning F_t into subsets, and then bit-stripping t with respect to only one subset of F_t at a time. If F_t is partitioned into n subsets, then t will be replaced by n different 3-valued vectors. Each of the n vectors will have more X's because they are targeting a smaller set of faults, and by including all of them in the test set, the overall fault coverage will remain the same.

The steps of the procedure for eliminating peak power violations are as follows:

Step 1: Identify and classify peak power violations. This was described in Sec. 2. Simulation is done to identify the peak power violations, and then classify them into scan capture problems, scan-in problems, scan-out problems, and order-dependent problems.

Step 2: Eliminate scan-in problems. Scan-in problems occur when a scan vector causes excessive switching activity in the circuit as it is scanned in. This can happen if there are a lot of transitions in the scan vector. If there is a scan-in problem for scan vector t, then bit-stripping is done to introduce X's into t. The X's are then reassigned to new values that result in fewer transitions (this will be described in detail in Sec. 4.1). Note that reassigning the X's in t to eliminate a scan-in problem may cause a scan capture, scan-out, or order dependent problem because t is now a different vector. So simulation must be done to check if t now has a scan capture, scan-out, or order dependent problem.

Step 3: Eliminate scan capture problems. Scan capture problems occur because a scan vector may place the circuit into a state that would not occur during normal functional operation. When the output response is captured in the scan chain, an excessive number of flip-flop transitions may occur which can

cause a peak power violation. Scan capture problems are much less likely than scan-in and scan-out problems because the number of transitions during scan capture will tend to be similar to what occurs during functional operation. In general, scan capture problems will be a rare occurrence. If there is a scan capture problem, then bit-stripping is done to introduce X's, and then the X's are reassigned to new values to reduce peak power during scan capture without causing a scan-in problem (this will be described in detail in Sec. 4.2). Simulation must be done to check if the modified scan vector now has a scan-out or order-dependent problem.

Step 4: Eliminate scan-out problems. Scan-out problems occur when the output response of a scan vector causes excessive switching activity in the circuit as it is scanned out. This can happen if there are a lot of transitions in the output response of a scan vector. To reduce the number of transitions in the output response of a scan vector, bit-stripping is done to introduce X's in the input, and then the X's are reassigned to new values that result in fewer transitions in the output response without causing a scan-in or scan capture problem (this will be described in detail in Sec. 4.3). After the X's are reassigned, simulation must be done to check if the modified scan vector now causes any order dependent problems.

Step 5: Eliminate order dependent problems. Once all the scan capture, scan-in, and scan-out problems have been eliminated, the last step is to eliminate any order dependent problems that may exist. Order dependent problems occur for a pair of vectors where shifting in test vector t_i concurrently with shifting out the output response of test vector t_{i-1} results in a peak power violation. First an attempt is made to eliminate an order dependent problem by simply moving scan vector t_i so that it comes after a different scan vector whose output response has fewer transitions and moving test vector t_{i-1} so that it comes before a scan vector that has fewer transitions. If this does not eliminate the problem, then a dummy vector is inserted in between test vector t_{i-1} and t_i to solve the problem. If it is important not to increase the size of the test set by adding a dummy vector, then another option is to reduce the number of transitions in either test vector t_i or the output response of test vector t_{i-1} by modifying the vectors in the same way as for solving scan-in or scan-out problems.

This gives an overview of the procedure for eliminating peak power violations. Note that the number of scan vectors in the test set may be slightly increased by this procedure. If the peak power limit is too low, the procedure is not guaranteed to be able to eliminate all peak power violations. It is possible that a fault cannot be detected during scan testing without causing a peak power violation if that fault required a large percentage of the scan elements to be at a specified value in order to be detected. If such a fault does exist, peak power violations can only be eliminated by dropping coverage for that fault. Otherwise, the procedure will eliminate all peak power violations while preserving fault coverage.

4. Reassigning Input Values to Eliminate Peak Power Problems

This section describes how the X's are reassigned to eliminate each of the types of peak power problems that can occur.

4.1 Eliminating Scan-In Problems

The first class of peak power problems that are eliminated are the scan-in problems. A scan-in problem occurs because a scan vector has too many transitions in it resulting in excessive switching activity as the vector is scanned in. To eliminate scan-in problems, the number of transitions in the vector needs to be reduced by reassigning input values. This is done by first bit-stripping the scan vector to introduce X's, and then doing a *minimum transition fill (MT-fill)* of the X's. MT-fill involves filling strings of X's with the same value to minimize the number of transitions. For example, when filling the vector 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 vector, 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 with either 0 or 1.

Figure 3 shows an example of reducing the number of transitions in the original vector by first bit-stripping and then doing an MT-fill of the X's. After doing MT-fill, simulation is done to check if the resulting vector still causes a peak power violation when it is scanned in. If there is still peak power violation, then *reverse bit-stripping* is performed which involves doing bit-stripping again, but processing the inputs in the reverse order from the original bit-stripping. This will tend to introduce a different set of X's into the vector. After reverse bit-stripping, MT-fill is done again, and a check is made to see if there is still a peak power violation. If there is still a peak power violation, then the set of faults,

F_t, that the scan vector, t, is targeting is partitioned into subsets so that bit-stripping will produce more X's. The scan vector t is bit-stripped with respect to each subset to produce a multiplicity of scan vectors each of which has more X's than t while together providing the same fault coverage as t. If the original set of faults, F_t, is small enough, then bit-stripping can be done with respect to each individual fault in F_t. Otherwise, F_t can be randomly divided into two subsets, and then repeatedly divided as necessary. The number of additional scan vectors that results from partitioning the fault set can be reduced by static compaction. The static compaction procedure can be constrained so that the number of transitions in the merged vector never exceeds some threshold [Sankaralingam 00]. For example, suppose that bit-stripping scan vector t resulted in a vector that had 100 transitions in the specified bits (the X's are assumed to be filled with MT-fill). However, by partitioning F_t into 8 subsets, the resulting 8 vectors after bit-stripping all have fewer than 50 transitions in the specified bits. At this point, static compaction can be used to merge some of the 8 vectors back together under the constraint that the resulting vectors after merging still have no more than 50 transitions in the specified bits.

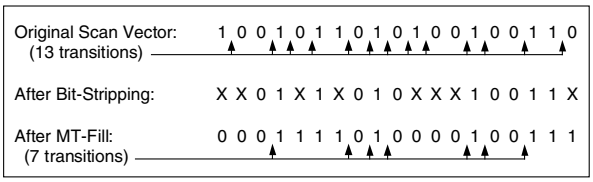

Figure 3. Example: Eliminating Scan-In Problem

4.2 Eliminating Scan Capture Problems

The second class of peak power problems that are eliminated are the scan capture problems. A scan capture problem occurs because an excessive number of flip-flops transition after scan capture. The flip-flops that transition after scan capture are those for which the input value differs from the output value. Given a scan vector, fault-free simulation can be done to determine its corresponding output response vector. To eliminate scan capture problems, the number of input/output differences in the vector needs to be reduced by reassigning input values. This is done by first bit-stripping the scan vector to introduce X's. The bit positions with X's will be referred to as "free inputs" because they can be freely reassigned without affecting the fault coverage. The rest of the bit positions, which do not have X's, are "fixed inputs." Once the set of free and fixed inputs has been determined via bit-stripping, we go back to original fully specified scan vector as our starting point. Fig. 4 shows an example. The goal now is to reassign some of the free inputs so that the number of input/output differences is reduced. A hill climbing approach can be used. We identify one free input whose value is different from the corresponding output. The value of the free input is complemented. Fault-free simulation is done on the altered input vector to obtain the new output response. A check is made to see if the number of input/output differences has decreased. If so, then the change is kept, if not then it is undone. This is repeated for all the free inputs. While this only explores a subset of the exponential number of possible free input assignments, it has the advantage of tending to minimize the number of changes to the original scan vector and thereby reducing the chance of creating a scan-in problem (since the original scan vector did not have a scan-in problem). After the inputs are reassigned in this manner, simulation is done to see if the scan capture problem has been eliminated and no scan-in problem has been created. If not, then reverse bit stripping can be done to identify a different set of free inputs and the procedure can be repeated. If still no solution, then bit-stripping with partitioning the fault set can be used to further increase the number of free inputs until a solution is found.

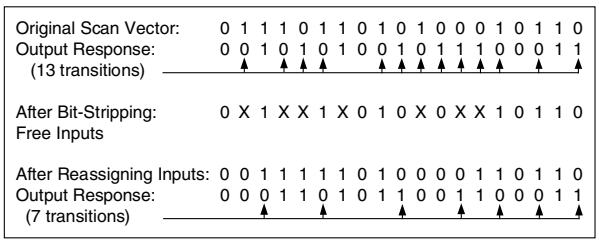

Figure 4. Example: Eliminating Scan Capture Problem

4.3 Eliminating Scan-Out Problems

The third class of peak power problems that are eliminated are the scan-out problems. A scan-out problem occurs because the output response of a scan vector has too many transitions in it resulting in excessive switching activity as it is scanned out. To eliminate scan-out problems, the number of transitions in the scan vector's output response needs to be reduced by reassigning input values. This is done by first bit-stripping the scan vector to introduce X's and thereby identify the set of "free inputs" whose value can be changed without affecting the fault coverage. Once the set of free and fixed inputs has been determined via bit-stripping, we go back to original fully specified scan vector. The goal now is to reassign some of the free inputs so that the number of transitions in the output response is reduced. Fig. 5 shows an example.

A key component that we use for reassigning the inputs is a line justification procedure like the one used in PODEM [Goel 81] except with a modification to the controllability cost function. Normally a cost of 1 is assigned for controlling each of the primary inputs to either a logic 0 or logic 1, and then the circuit is traversed from the primary inputs to the primary outputs to compute the controllability cost function at each individual node based on these initial values for the primary inputs. In our case, since we cannot change the value of the fixed inputs, if we have a fixed input at a logic 1 (0), then we assign a cost of infinity (zero) for controlling that input to a logic 0 (1), and assign a cost of zero (infinity) for controlling that input to a logic 1 (0). For the free inputs, we assign a cost of 1 for controlling the input to its original value, and a cost of 10 for controlling the input to the complement of its original value. The reason for this is to bias the procedure towards reducing the number of changes to the original scan vector in order to reduce the chance of creating a new scan-in or scan capture problem (since the original scan vector did not have any scan-in or scan capture problem to start with).

We use a hill climbing procedure for reducing the number of transitions in the output response. We look at the output response and identify all the bit positions where the controllability value for complementing the output value is less than infinity. These are the candidate outputs whose value can be changed by reassigning free inputs. We identify the candidate output with the lowest controllability value where complementing its value would eliminate a transition. The line justification procedure is used to find an assignment of inputs that complements this output's value. All the input assignments to fixed inputs made by the line justification procedure will be the same as their original value by definition of our controllability cost function. Some of the input assignments to the free inputs will be different than their original value and some may be the same. After the input assignments have been made, fault-free simulation is done to obtain the new output response. If the number of transitions in the output response decreases, then the input changes are kept, otherwise they are undone. If the input changes are kept, then all free inputs that required assignments to justify the output (regardless of whether they were the same or different than their original values) become fixed inputs. The controllability values are then recalculated and the process repeats until a point is reached where no more candidate outputs can be reassigned to reduce the number of transitions. At this point, simulation is done to see if the scan-out problem has been eliminated while not creating any scan-in or scan capture problems. If not, then reverse bit stripping can be done to identify a different set of free inputs and the procedure can be repeated. If still no solution, then bit-stripping with partitioning the fault set can be used to further increase the number of free inputs.

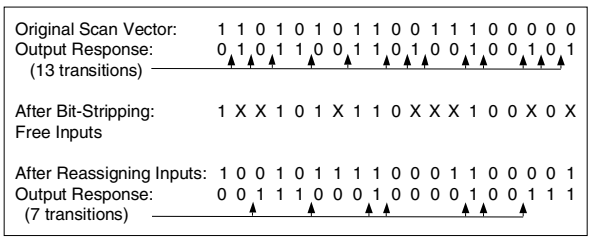

Figure 5. Example: Eliminating Scan-Out Problem

5. Experimental Results

Experiments were performed on the largest ISCAS 89 benchmark circuits. The primary inputs and primary outputs were included in the scan chain. A test set was generated for each circuit using a commercial ATPG tool with maximum compression and MT-filling of the X's. For each circuit, scan testing was simulated using the test set. For each new bit scanned in, the circuit was evaluated to calculate the number of internal gates that change state. Each gate transition was weighted by the number of fanouts of the gate. After all the bits for one scan vector are scanned in, the scan capture cycle is simulated. The output response is then shifted out as the next vector is scanned in. This simulation process was used to find the maximum number of weighted transitions in any cycle during scan testing.

The proposed procedure was then used to modify the scan vector test set to reduce the maximum peak power by 10% (shown in Table 1) and by 20% (shown in Table 2). The number of scan vectors that cause each type of peak power violation are shown in the tables for the respective maximum peak power limits. In our experiments, we did not encounter any scan capture problems. All of the peak power violations occurred during scan shifting. The proposed procedure modified the scan vectors to eliminate the scan-in and scan-out problems, and then reordered or added dummy vectors to eliminate the scan dependent problems. The original number of vectors in the test set is shown in each table followed by the final number of vectors after using the proposed procedure. There is a slight increase in the number of vectors due to adding dummy vectors and bit-stripping vectors with respect to partitioned fault sets (as described in Sec. 4).

As can be seen from the results, the procedure is able to significantly reduce peak power without the use of DFT hardware. The only cost is a small increase in the test set size.

Table 1. Results for Reducing Peak Power by 10%

Circuit		Number of Scan Vectors with Each Type of Problem				Number of Vectors	
Name	Scan Cells	Scan-In	Scan-Out	Scan Capture	Order Dependent	Original	Final
s9234	247	5	4	0	27	154	156
s13207	700	2	8	0	16	240	246
s15850	611	2	11	0	0	118	120
s38417	1664	0	0	0	2	96	98

Table 2. Results for Reducing Peak Power by 20%

Circuit		Number of Scan Vectors with Each Type of Problem				Number of Vectors	
Name	Scan Cells	Scan-In	Scan-Out	Scan Capture	Order Dependent	Original	Final
s9234	247	7	76	0	55	154	164
s13207	700	2	76	0	59	240	247
s15850	611	1	23	0	42	118	128
S38417	1664	8	3	0	4	96	100

6. Conclusions

The scan vector modification procedure described here can be used to keep the peak power during scan testing under a specified limit. The user can adjust the specified limit as necessary to avoid failures due to excessive peak power during scan testing that would not occur during normal functional operation. The procedure may increase the test set size slightly.

Note that the proposed procedure can be used for any set of fully specified scan vectors. There is no requirement for any special ATPG process. Normally during ATPG, random filling of the X's is used to try to get each scan vector to cover more faults. A simple approach to reduce power is to use MT-filling of the X's. The proposed procedure can be used in either case to reduce peak power. The experimental results showed that even with MT-filling of the X's during ATPG, the proposed procedure could significantly reduce peak power beyond that.

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 Technology 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.

[Chandra 01] Chandra, A., and K. Chakrabarty, "Combining Low-Power Scan Testing and Test Data Compression for System-on-a-Chip," *Proc. of Design Automation Conf.*, pp. 166-169, 2001.

[Corno 99] Corno, F., M. Rebaudengo, M. Sonza Reorda, M. Violante, "On Reducing the Peak Power Consumption of Test Sequences," *European Conf. on Circuit Theory & Design*, pp. 247-250, 1999.

[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 Comp.-Aided Design*, Vol. 17, No. 12, pp. 1325-1333, Dec. 1998.

[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 99a] 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.

[Girard 99b] Girard, P., L. Guiller, C. Landrault, and S. Pravossoudovitch "Circuit Partitioning for Low Power BIST Design with Minimized Peak Power Consumption," *Proc. of Asian Test Symposium*, pp. 89-94, 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.

[Lee 00] Lee, K.-J., T.-C. Huang, and J.-J. Chen, "Peak-Power Reduction for Multiple-Scan Circuits during Test Application," *Proc. of Asian Test Symposium*, pp. 453-458, 2000.

[Sankaralingam 00] Sankaralingam, R., R.R. Oruganti, and N.A. Touba, "Static Compaction Techniques to Control Scan Vector Power Dissipation", *Proc. of VLSI Test Symp.*, pp. 35-40, 2000.

[Sankaralingam 01] Sankaralingam, R., B. Pouya, and N.A. Touba, "Reducing Power Dissipation During Test Using Scan Chain Disable", *Proc. of VLSI Test Symposium*, pp. 319-324, 2001.

[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 International Test Conference*, 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.

[Whetsel 00] Whetsel, L., "Adapting Scan Architectures for Low Power Operation," *Proc. of Int. Test Conf.*, pp. 863-872, 2000.