ATPG for Heat Dissipation Minimization During Test Application
Seongmoon Wang and Sandeep K. Gupta

Abstract—A new automatic test pattern generator (ATPG) algorithm is proposed that reduces switching activity (between successive test vectors) during test application. The main objective is to prevent safe and inexpensive testing of low power circuits and to reduce the number of transitions between test vectors. Three new cost functions, namely, transition controllability, observability, and test generation cost, have been defined. It has been shown for a fanout-free circuit under test, that the minimum number of transitions required to test a given fault at a fault is the minimum number of transitions required to test a given fault at a fault. The proposed algorithm has been implemented and the generated tests are compared with those generated by a standard PODEM implementation for the larger ISCAS88 benchmark circuits. The results clearly demonstrate that the tests generated using the proposed ATPG can decrease the average number of transitions between successive test vectors by a factor of 2 to 25.

Index Terms—Combinational ATPG, switching activity, heat dissipation, PODEM, testing.

1 INTRODUCTION
ATTAINMENT of high fault coverage has traditionally been the main objective of the test development techniques. As these techniques have matured and this objective has been attained, other objectives have become important. Decreasing test application time and test data volume are other such objectives being pursued. We believe that reducing heat dissipated during test application will rapidly become another objective of the test-development techniques.

1.1 Motivation
The correlation between consecutive test vectors applied to a circuit, unless these test vectors are ordered specifically to increase correlation, is often significantly lower than that between consecutive vectors applied to the circuit during its normal operation. The fact that a significant correlation exists between consecutive vectors during the normal operation of a circuit is what has motivated several architectural concepts, such as cache memory, and is central to their effectiveness. This is even more true for the high-speed systems that process signals such as audio and video signals—these inputs to most of whose modules change relatively slowly over time. In contrast, the correlation between consecutive test vectors generated by an automatic test pattern generator (ATPG) is very low, since it is generated for given target fault without any consideration of the previous vector in the test sequence. Hence, the switching activity in a circuit can be significantly higher during test application than during its normal operation. The use of design for testability features, such as scan and built-in self-test (BIST), further exacerbates the problem. The use of scan destroys the correlation that typically exists between successive states of a finite state machine by allowing the application of any desired values to the state flip-flops. In BIST, the application of a pseudo-random sequence, whose consecutive vectors are proven to have low correlation, greatly increases switching activity. Excessive switching activity due to low correlation between consecutive test vectors can cause several problems.

First, since heat dissipation in a CMOS circuit is proportional to switching activity, a circuit under test (CUT) can be permanently damaged due to excessive heat dissipation if switching activity in the circuit during test application is much higher than that during its normal operation. The heat dissipated during test application is already influencing the design of test methodologies for practical circuits. For example, it is reported in [15] that one of their major considerations in test scheduling is the fact that heat dissipated during test application is typically significantly higher than that during their circuits' normal operation (sometimes 100-200 percent higher). The seriousness of excessive heat dissipation during test application is tied to trends such as circuit miniaturization for portability and high performance (smaller chips are being placed closer, decreasing interconnect delays). This is achieved by using circuit designs that decrease power dissipation and reducing the package size to aggressively match the average heat dissipation during the circuit's normal operation [13]. In order to ensure nondestructive testing of such a circuit, it is necessary to remove any excessive heat generated during test using special cooling equipment or apply test vectors which cause switching activity that is comparable to that during the circuit's normal operation. Though the use of special cooling equipment can remove heat dissipated during test, it cannot solve other problems caused by excessive switching activity, which are described in following.

Furthermore, removing heat by using special cooling equipment during test application becomes increasingly difficult and costly as tests are applied at higher levels of circuit integration, such as board and system levels.

Second, it has been observed that metal migration or electromigration of conductors and subsequent failure of circuits [12]. Since temperature and current density are major factors that determine electromigration rate, elevated temperature and current density caused by excessive switching activity during test application can severely influence the reliability of circuits under test. This is even more severe in circuits equipped with BIST since such circuits are tested frequently.

Finally, to test a bare die, power must be supplied during the period of test through probes which typically have higher inductance than power and ground pins of a circuit package. Hence, the bare die under test will experience higher power-ground noise which is given by \( LdI/dt \), where \( L \) is the inductance of power and ground line and \( dI/dt \) is the rate of change of current flowing in power and ground lines. Excessive power-ground noise can erroneously change the logic state of circuit lines causing some good die to fail the test. Since the tests generated by the proposed ATPG reduce switching activity, magnitude of current and \( dI/dt \) can be reduced during test application, thereby preventing such unnecessary loss of yield due to the limitation of probing.

Why, then, has the heat dissipation problem during testing not been considered in the past? There are two recent developments that are only beginning to bring this issue to light. In the past, the tests were typically applied at rates much lower than a circuit's normal clock rate (since only the coverage of stuck-at faults was deemed to be important and slow testers provided an inexpensive way of testing). However, in recent years, aggressive timing has made it essential for the tests to identify slow chips via delay testing. This is especially important for the growing number of circuits that are being manufactured for use in MCUs. For such circuits, delay testing is not only highly desired but almost imperative—a fact reflected in the extensive demand for performance-certified die [4], [14]. Consequently, circuits are now tested at higher clock rates—if possible, at the circuit's normal clock rate...
(cable at-speed testing). Consequently, heat dissipation during test application is on the rise and is fast becoming a problem that requires close attention.

This paper describes a new ATPG algorithm that generates tests for stuck-at as well as transition delay faults which minimize switching activity in circuits under test during their application.

1.2 Proposed ATPG and Its Application

It should be noted that the objective of the proposed ATPG is to ensure that switching activity during test application is low enough to ensure safe and nondestructive at-speed testing of devices under test. The tests generated by this ATPG can be used to test at-speed chips and boards, without running the risk of damaging devices under test due to excessive switching activity; the reduced power consumption during testing is, by itself, of little consequence since the test time is an insignificant part of a circuit's operational life.

The proposed ATPG assumes that the circuit under test is combinational. This ATPG is, however, applicable to several different types of sequential circuits. First, the test vectors generated by the proposed ATPG can be used in test-self-test scenarios where the test sequence is either stored or generated on-chip, using either functional or dedicated BIST circuitry. For example, consider the testing of a microprocessor. The proposed technique can be used to generate tests for an ALU and the generated test sequence can be applied by the processor, using a combination of stored instructions and test data. Note that, in such cases, one test can be applied per clock and heat dissipation would be very close to what is reported by the proposed ATPG.

Second, several large sequential circuits, especially those used for digital signal and image processing, are pipelined and can be treated as combinational circuits for test generation [8]. The proposed technique can be applied to such circuits without any modification. Many more circuits may not be strict pipelines, but can be modified into pipelines by the use of a very small number of scan flip-flops [8]. The proposed ATPG can then easily be employed, especially if parallel scan, discussed next, is employed.

Third, the use of parallel scan [2] is being proposed to obviate the problem of excessive test application time that is encountered in testing via traditional (serial) scan [9]. The test application methodology in parallel scan is very similar to that for combinational circuits, since the test data is loaded in parallel into scan flip-flops by multiplexing the I/O pins. Experimental results (see Section 5) clearly show that the tests generated by the proposed ATPG can obtain almost as large a decrease in heat dissipation in sequential circuits with parallel scan as they do when the circuit is assumed to be combinational.

Previous research in this area is represented by [15], where a maximum allowable dissipation is used as a constraint during scheduling of self-test. Also, a methodology to reorder test vectors generated by an ATPG to reduce heat dissipation during their application via scan is presented in [3].

2 Preliminaries

In the following, the circuit under test (CUT) is assumed to be a CMOS digital circuit. Heat dissipation in CMOS can be divided into static and dynamic. Static dissipation is due to leakage current that has small magnitude in digital CMOS circuits. Hence, for such circuits, the dynamic dissipation is the dominant term. Dynamic dissipation occurs at a node when it switches from 0 → 1 or from 1 → 0 and is typically divided into two components caused by short-circuit current and charge/discharge current, respectively. The former is caused by a current pulse that flows from power supply to ground when both n- and p-transistors are simultaneously on during switching, and is negligible in high-speed circuits where both n- and p-transistors are simultaneously on for very short periods. The charge/discharge current is the current that charges and discharges the capacitive load on the output of a gate and, in general, dominates dynamic dissipation. The dynamic power dissipation in the circuit is given by

\[ \sum \frac{1}{C_L(t)} \frac{dV(t)}{dt} \]  

where \( C_L(t) \) is the load capacitance at line \( t \) of the CUT, \( s(t) \) is the frequency of switching of the line \( t \), and \( V_{dd} \) is the power supply voltage. Other quantities being constant for a given circuit, tests generated to minimize the frequency of switching at circuit lines during test application will minimize heat dissipated during testing.

The zero delay model is used, almost exclusively, by test generators and fault simulators for stuck-at faults and is also used by the proposed algorithm. Under the zero delay model, no hazards are considered at the circuit lines and the dissipation is assumed to be mainly due to 0 → 1 and 1 → 0 transitions at the circuit lines. The use of zero delay model is justified by the observation that the power dissipation estimated under this model has a high correlation with that under the general delay model [10]. Additionally, the experiments conducted by us demonstrate that the tests generated by the proposed ATPG decrease the number of hazards in the same proportion as they reduce transitions.

Let \( T_0 \) and \( T_1 \) be consecutive test vectors applied to the CUT during testing. Let \( \text{paul}(t) \) and \( \text{null}(t) \) be the values implied at a line \( t \) of the circuit by the vectors \( T_0 \) and \( T_1 \) respectively. The amount of heat dissipated due to the application of the test vector \( T_i \) is proportional to

\[ \sum \{ \text{null}(t) \oplus \text{paul}(t) \} C_L(t) \]  

The objective of the proposed ATPG is to generate a test sequence such that heat dissipated during test application is minimized. It is assumed that the tests are applied in the order in which they are generated by the ATPG. Reducing switching activity would reduce heat dissipation caused by short-circuit, as well as charge/discharge current, because both these currents occur only during transitions. Clearly, the test sequences generated must provide high fault coverage.

At the heart of the proposed methodology are three cost functions, obtained by generalizing and modifying the traditional cost functions, such as controllability and observability, and described next.

3 Cost Functions

Three cost functions, namely,

1. transition controllability cost,
2. transition observability cost, and
3. transition test generation cost,

are defined for use by the proposed ATPG. These cost functions are computed for every circuit line, taking into consideration the values implied at all circuit lines by the previous vector in the test sequence, i.e., \( \text{paul}(t) \). The transition controllability cost at a line indicates the minimum weighted transitions required to set that line to a desired logic value. The transition observability cost at a line indicates the minimum weighted transitions required to propagate an error value (0 or 1) from that line to a primary output. The transition test generation cost for a fault, which is the sum of transition controllability and transition observability costs, indicates the minimum weighted transitions that will occur to detect the fault by the next test vector.

In general, a CUT is assumed to be comprised of AND, OR, NAND, NOR, NOT gates and buffers. A gate \( g \) is characterized by its controlling value \( c \) and its inversion \( i \). If a gate is inverting, such as NAND, NOR, or NOT, then the inversion \( i \) of the gate is said to
Fig. 1: A node $g$ in the CUT.

be 1, otherwise, $i$ is said to be 0. If a value $c$, when applied to an input of a gate, determines the value at the output of the gate regardless of the values applied to its other inputs, then the value is said to be the controlling value of the gate. For example, the controlling value of AND and NAND gates is 0. Note that, if any input to a gate has a value $c$, then the value at the gate output is $c \oplus i$. For the part of an example CUT shown in Fig. 1, let $i$ be the output of an $n$-input gate $g$ and let $I_1, I_2, \ldots, I_n$ be the inputs of $g$. Furthermore, assume that the gate $g$ has a fanout of $m(i)$ with $f_1, f_2, \ldots, f_m$ as its fanout branches.

### 3.1 Transition Controllability Cost

Consider the objective of setting a line $i$ in a CUT to value $v$, where $v$ is either a 1 or a 0. Consider the fanin cone containing gates that feed the line $i$ directly or indirectly, including line $i$ itself. Let $TCO(i)$ be the minimum weighted transitions in the CUT required to assign the value $v$ to the line $i$. If $prout(i) = v$, then $TCO(i) = 0$, that is, no transition is required in the fanin cone to assign the desired value. However, if $prout(i) \neq v$, the fanin cone will have at least one transition—at line $i$. The actual number of transitions required to set a line $i$ to a value $v$ is computed by first assigning $i$ a $TCO(i)$ value for all primary inputs. At a primary input $i$, $TCO(i)$ is assigned the value 0 if $prout(i) = v$; otherwise, it is assigned the value $C(i)$, where $C(i)$ is the capacitive load on the primary input $i$.

Starting at the primary inputs, the values of the two transition controllability cost functions, $TCo(i)$ and $TCO(i)$, are computed for every line $i$ via a breadth-first traversal of the circuit. These two cost functions reflect minimum weighted transitions required to set line $i$ to a 1 and a 0, respectively, and are used to guide the backtrace procedure in the proposed ATPG along paths that minimize the number of transitions in the circuit. The objective to set $i$ to $c \oplus i$ can be achieved by setting any one of the gate inputs to $c$. On the other hand, if the objective is to set $i$ to $\bar{c} \oplus i$, then every input line of the gate must be set to $\bar{c}$. $TCO(i)$ is, hence, given by

$$TCO(i) = \begin{cases} 0 & \text{if } i = \text{primary input} \\ C(i) + \min\{TCO(j)\} + K(i) & \text{if } v = c \oplus i \\ C(i) + \sum\text{TCO(j)} + K(i) & \text{if } c = c \oplus i. \end{cases}$$

where $j$ are all the inputs of a gate $g$ with output $i$, and $K(i)$ is a positive constant when $m(i) > 1$, 0 when $m(i) = 1$. Finally, $f_j$ is a fanout branch with fanout stem $i$ (Fig. 1), then $TCO(f_j) = TCO(i)$.

Let the output of a gate $g$ be a fanout stem driving $m(i)$ fanout branches $f_1, f_2, \ldots, f_m$ (Fig. 1). If $f_j$ has a transition, it will cause a transition at all its branches. Furthermore, the output lines of some of the gates $g_1, g_2, \ldots, g_{m(i)}$ that are driven by $f_1, f_2, \ldots, f_m$ may also have transitions, depending on the values at their other inputs. The transitions at these lines may cause transitions at the output lines of the gates that are driven by them. In an extreme case, a large number of circuit lines may have transitions, even when the fanout of $i$ is small. Determining exactly how many other lines can have transitions is difficult during the calculation of the cost functions, because the values at the other inputs of the gates $g_1, g_2, \ldots, g_{m(i)}$ are not known at that time. $K(i)$ is introduced in the above expressions for $TCO(i)$ as a correction term to account for such potential transitions. Since a fixed $K(i)$ is used for all fanout stems in the circuit, $TCO(i)$ is an approximate measure of the weighted transitions that would occur in the circuit to set the line $i$ to the value $v$.

When a circuit is fanout-free, any line justification can be performed independently of other justifications. Thus, if an input line $i$ of a gate with output $l$ has smaller $TCO(l)$ than its other inputs, the line $l$ can be set to $c \oplus i$ by setting $i$ to $c$ without any conflict.

**Lemma 1.** In a fanout-free circuit, $TCO(l)$ is the minimum weighted transitions required to set the line $l$ to $v$.

**Proof.** This result can be proven by induction. Initially, a primary input whose previous value is different from the new value has one transition and a primary input whose previous value is the same as the new value stays constant (zero transition). Hence, in general, a primary input $i$ has $TCo(i)$ transitions. Assume that all lines $h$ in the level $k$ have $TCo(h)$ transitions. Consider a gate at a level $k + 1$ with output $l$ and inputs $I_1, I_2, \ldots, I_n$. When $v = \bar{c} \oplus i$, every input line, $I_1, I_2, \ldots, I_n$, should be set to $\bar{c}$. Thus, the number of transitions at line $l$ is

$$C(i) + \sum\text{TCO(i)} = TCO(l).$$

The following example illustrates how $TCo(i)$ values are computed.

**Example 1.** Fig. 2 shows a part of an example circuit where $p_1, p_2$, and $p_3$ are primary inputs and $q_1, q_2$, and $q_3$ are fanout branches of $l$. The number in parenthesis following each line name is $prout(i)$, the value implied at the line by $TCo(i)$. Let us calculate the transition controllability costs for each line.

Assuming $K(i)$ to be 0, the transition controllability cost $TCo(l)$ is 0 for all lines except $p_0$, because their previous values are 1; $TCo(p_0)$ is 9 for a similar reason. Assuming $C(i)$ to be $m(i)$, since the lines $p_1, p_2$, and $p_3$ have fanout 1, we obtain $TCo(p_0) = TCo(l) - 1$, and $TCo(l) = m(l) + \min(TCo(l))$, $TCo(p_0) = 2$. Also, $TCo(l) = m(l) + TCo(l) + TCo(l) = 5$, since $m(l) = 3$. Finally, $l$, $q_1$, and $q_2$ have the same transition controllability costs as $l$, because $l$, $q_0$, and $q_3$ are the fanout branches of $l$.

### 3.2 Transition Observability Cost

In the PODEM ATTC [5], [11], at the beginning of each test generation step, a vector $T_i$ is initialized to $x_1, x_2, \ldots, x_n$ where $x_n$ represents either an unknown or a don’t care value. During the test generation, the values are specified at the inputs of the circuit, one input at a time. At any intermediate step, all gates whose output values are currently $x$ but that have error value(s) at one or more of their inputs are said to belong to the $D$-frontier [5], [11]. In order to detect the target fault, the error(s) at the input(s) of at least one gate that belongs to the D-frontier should be propagated to one of the circuit primary outputs, with minimum weighted transitions. In each step of the test generation, the transition observability cost func-
Fig. 2. An example circuit.

The TOn(l) is computed for every line l and is used to select a gate from the D-frontier. TOn(l) is the minimum weighted transitions required to propagate the error at a line l, whose good circuit value is v, to a primary output.

In the circuit of Fig. 1, let g be a gate that currently belongs to the D-frontier. Assume that we want to propagate an error from its input i to its output l. This objective can be achieved by setting all inputs of g, except \( i \), to \( \bar{v} \). A transition may or may not occur at l depending on the current value v at i and the previous value of l. If \( posl(l) \neq v \oplus i \), then l will have a transition when v is propagated to l; otherwise, l will not have a transition. This fact is captured in a term \( q \), which is assigned 0 when \( posl(l) = v \oplus i \) and \( C_{l}(i) \) when \( posl(l) = v \oplus \bar{i} \), in the following expression to compute the transition observability of the input \( i \) of the gate g:

\[
TOn(l) = q + K(l) + TOn(v) + \sum_{l \in \mathcal{L}} TOn(l),
\]

where \( i \) are all inputs of the gate g except \( i \), \( v' = v \oplus i \), and K(l) is as defined earlier.

\[
TOn(i) = \min_{q \in \mathcal{E}} \{ TOn(f_{i}) \},
\]

where \( f_{i} \) are fanout branches of stem l. Finally, if l is a primary output, then TOn(l) = 0. While TOn(l) is an estimate if the circuit has fanouts, when a circuit is fanout-free, \( \mathcal{L} \) can be assigned to any input of l which is currently assigned x without any conflict and there is always only one gate on the D-frontier. This leads to the following result.

**Lemma 2.** In a fanout-free circuit, TOn(l) (TOn(i)) is the minimum weighted transitions required to propagate the fault effect \( D \) (\( T \)) from line l to the primary output.

### 3.3 Target Fault Selection

The proposed ATPG is described for the stuck-at-fault model. The ATPG for transition delay fault model is similar and hence it is not described.) Assuming the existence of a procedure that can generate a vector that can detect a given targeted fault with a minimum number of transitions for a given primary vector \( T_{i} \), we require a criterion to select a suitable target fault.

In order to generate a vector to detect a stuck-at-\( \bar{v} \) (s-a-\( \bar{v} \)) at line l, first, the fault should be activated by setting \( i \) to v. The activated fault effect should then be propagated to one or more primary outputs. The function TCOn(l) is a measure of the former and TOn(l) is a measure of the latter. In each step of the test generation, the transition test generation cost, TCOn(l), is calculated for every remaining fault in the fault list and is used to select the next target fault. TCOn(l) for the stuck-at-\( \bar{v} \) fault at line l is given as:

\[
TCOn(l) = TCOn(l) + TOn(l).
\]

**Theorem 1.** In a fanout-free circuit, TCOn(l) of a line l is the minimum weighted transitions required to activate the stuck-at-\( \bar{v} \) fault at line l and propagate the fault effect to the primary output.

**Proof.** Follows from Lemmas 1 and 2.

**Note.** That the selection of a target fault using TCOn(l) value is a greedy strategy. Other strategies are currently being investigated. Also note that this function is approximate for circuits with fanouts because the same value of \( K(l) \) is used for every fanout stem in CUT; a more accurate cost function can be used at the expense of increased cost computation time.

### 4 Proposed ATPG Algorithm

The proposed ATPG algorithm, referred to as the proposed PODEM, is based on the PODEM algorithm [5]. A detailed description of this algorithm can also be found in [1]. The proposed PODEM initializes the fault list to contain all checkpoint faults [1]. At each step, all three transition cost functions are computed for all lines, and the fault in the fault list that has the lowest test generation cost is selected as the target fault. Proposed PODEM then generates a test vector for the target fault. All routines used in the proposed PODEM are identical to those used by the original PODEM, except that the proposed PODEM uses the transition controllability costs to guide the backtrace and the transition observability costs to select gates from the D-frontier for fault effect propagation. In order to achieve this, two procedures in the original PODEM, \textit{Backtrace}() and \textit{Objectiv()}, are modified.

The procedure \textit{Backtrace()}, which is a modified version of the procedure \textit{Backtrace()} of the original PODEM, determines the path to backtrace using the transition controllability cost. For example, consider the R-input and J gate shown in Fig. 1. If the current objective is to assign a 0 to its output l, \textit{Backtrace()} selects that input \( i \) which has the lowest TOn(l). Then, it continues to backtrace in a similar manner, until it reaches a primary input and assigns the desired value to the primary input. The procedure \textit{Objectiv()}, the other routine that is modified, selects the gate that has minimum transition observability cost, among the gates that belong to the D-frontier, for fault effect propagation. Next, it uses \textit{Backtrace()} to assign noncontrolling values to all other inputs of the gate g.

If the test vector \( T_{i} \) generated by the proposed PODEM has any don't cares, the number of transitions depends on the assignment of 0 or 1 to the don't care bits. A new procedure, called \textit{don't_care()}, is developed to assign an appropriate value to each don't care bit of \( T_{i} \) to minimize the number of transitions. Assume that the primary inputs \( l_{1} \) and \( l_{2} \) of the circuit shown in Fig. 3 are assigned 0 and 1, respectively, by the vector \( T_{i} \), which sets the gate output l to 0. If the next vector \( T_{i} \) assigns a 1 to \( l_{1} \) and a don't care to \( l_{2} \), then if the line \( l_{2} \) retains its previous value (1), the gate's output will have 0 to 1 transition, which can cause several transitions. On the other hand, if \( l_{2} \) is assigned 0, the line l will not have a transition. Hence, in cases where an input \( i \) of a gate \( g \) changes its value from controlling value to noncontrolling value and another input \( l_{g} \) of \( g \), which is a primary input, is assigned a don't care value by the test generator, the input with don't care is assigned the gate's controlling value to avoid a transition at the gate output. Generally, a vector generated by an ATPG such as PODEM contains many don't care bits. Thus, the appropriate assignment of 0 or 1 to the don't care is very important. Our experimental results show that
the number of transitions can be reduced by up to 75 percent by using only the procedure \textit{don't care} [11], even without using the transition cost functions defined above.

The complete test generation algorithm for stuck-at faults is outlined below:

1. Cost functions are calculated for every line in the CUT and the fault that has the minimum transition test-generation cost, among the faults that belong to the fault list, is selected as the next target fault.

2. The proposed PODEM is used to generate a vector that detects the target fault with a minimum number of transitions. If the fault is found to be undetectable, then it is dropped from the fault list.

3. If the vector has any \textit{don't care} bits, then the procedure \textit{don't care} is used to assign appropriate binary values to these bits.

4. Faults detected by the vector are dropped from the fault list.

5. Steps 1 to 4 are repeated until no faults remain in the fault list.

The above ATPG algorithm has also been generalized to generate tests for transition delay faults with reduced transitions.

5 EXPERIMENTAL RESULTS

An existing version of PODEM was modified to implement the proposed PODEM. Procedures to compute the transition controllability, observability, and test generation costs for circuit lines were incorporated into the existing implementation. Note that these costs are all functions of the previous test vector applied to the circuit and, hence, must be recomputed at the beginning of each test generation step. Since the complexity of the cost computation is linear in the number of circuit lines, the overall impact of the repeated computation of the cost functions on the ATPG time complexity is small.

Table 1 compares the results obtained by using the original and the proposed versions of PODEM for the larger ISCAS 85 benchmark circuits. The original PODEM uses cost functions similar to those described in [6, 7] to select fault propagation objectives and to guide the backtrace procedure. Also, at each test generation step, the target fault is selected arbitrarily and any \textit{don't care} in the generated test vector are assigned randomly. In contrast, the proposed PODEM uses the proposed transition cost functions and
the don't care assignment procedure described above. Table 1 compares average numbers of weighted transitions, average numbers of lines with potential hazards, fault coverages, total numbers of vectors, average numbers of backtracks, and test generation times for these two algorithms. A backtrack limit of 200 was used in all experiments, and the run times reported are for a SUN 4/75 with 32 MBytes of memory. In this table, the numbers shown within parentheses are values normalized with reference to the data obtained by using the original PODEM implementation. Finally, the row labeled eff. reduc. reports the ratio between the number of transitions caused during the application of the entire test sequence obtained by using the proposed PODEM and that obtained by using the original PODEM.

It should be noted that the reduction in the average number of transitions, per vector, ranges from 52 percent to 96 percent. In other words, the average heat dissipation between successive test vectors generated by the proposed PODEM is a factor of 2 to 23 lower than that for the tests generated by the original implementation. It should also be noted that large reductions in the average number of transitions occurs for the larger circuits. Recall that, in all cases, the tests are applied in the order in which they are generated by the ATPG. Also recall that the transitions at circuit lines are weighted by their fanout values m(l), which is used as an approximation of C(l).

Even though our modified PODEM doesn't explicitly consider delay, the experimental results clearly show that the modified PODEM reduces the number of circuit lines where a hazard can potentially occur. The row labeled average # of hazards shows the total number of nodes which potentially can have hazards during the entire test application. Note that the number of lines that can potentially have hazards is decreased by a proportion similar to the number of transitions by the tests generated by the proposed ATPG.

The test sequences generated by both PODEM implementations obtain almost identical fault coverage for each circuit. Test sequences generated by the proposed ATPG are 3.88 to 4.2 times as long as those generated by the original PODEM. An extensive analysis of these implementations and variations obtained by combining their features shows that it is possible to trade-off the decrease in heat dissipation for a smaller increase in test length [11].

Note that the average number of backtracks is lower for the proposed ATPG for most circuits, even though the transition cost functions, which do not reflect the difficulty of observing or controlling the circuit lines, are used to guide the objective selection and backtrack procedures.

Extensive analysis shows that the additional time complexity, per test vector, of the proposed PODEM is small [11]. This is due to the fact that the proposed PODEM only requires additional cost computation, which has polynomial time complexity.

Additional experiments were performed to demonstrate that the test sequences generated by the proposed ATPG not only reduce heat dissipation while achieving high coverage of stuck-at faults, but also achieve high coverage of transition delay faults. This data indicates that the reduction in heat dissipation is not at the expense of the loss of coverage of unmodeled faults such as delay faults. To further study the effectiveness of the proposed ideas, we generalized the proposed ATPG (as well as the original one) to generate tests for transition delay faults. Experimental results (Table 2) show that the ideas proposed above are effective even to obtain high coverage of transition delay faults with greatly reduced number of transitions. Note that the decrease in the number of transitions is a little lower than that in the case of stuck-at faults. Nonetheless, the reductions are significant in magnitude (factor of two to 18). Hence, the empirical results clearly demonstrate that the proposed ATPG algorithm can significantly reduce the average and peak heat dissipation between successive vectors and achieve high fault coverage.

Next, we considered parallel scan implementations of ISCAS 89 sequential benchmark circuits. In parallel scan, the 1/O pins of a chip are used to control and observe data from the circuit flop-flops in parallel. Hence, if the number of flip-flops is lower than the number of available 1/O pins, test data is loaded into all flip-flops in a single clock cycle. Another clock cycle is then required to apply appropriate test data to primary inputs, requiring two clocks per test. In general, if a circuit has Nf flip-flops and NQ 1/O pins, then a total of (Nf/NQ) + 1 clock cycles are required to apply a test vector generated by a combinational ATPG. Experiments were performed with the tests generated by the two implementations (original and proposed ATPG) for the combinational parts of these circuits assuming parallel scan. The experimental results show that the tests generated by the proposed PODEM decrease the average number of transitions by a factor of two to five, without any significant increase in test length.

6 Conclusion

A new ATPG algorithm has been proposed that reduces average heat dissipation between successive test vectors during test application. The main objective is to permit safe and inexpensive testing of low power circuits and bare die that would otherwise require expensive heat removal equipment for testing at high speeds. Three new cost functions, namely transition controllability, observability, and test generation costs, have been defined. It has been shown that the transition test generation cost for a fault is the minimum number of transitions required to test a given stuck-at fault if the circuit under test is fanout free. This cost function is used to select a target fault while the other two functions are used to guide the backtrack and objective selection procedures of PODEM.

The proposed algorithm has been implemented and the generated tests are compared with those generated by a standard PODEM implementation for the larger ISCAS 85 benchmark circuits. The results clearly demonstrate that the tests generated using the proposed ATPG can decrease the average number of (weighted) transitions between successive test vectors by a factor of 2 to 23. It is interesting to note that large reductions are obtained for the larger circuits. The additional overhead due to recursive computation of cost functions is small and the fault coverage is the same as the original PODEM. The proposed algorithm has also been demonstrated to decrease the average number of transitions by 50-80 percent for combinational as well as parallel scan versions of sequential benchmark (ISCAS 89) circuits.

The proposed algorithm has also been successfully generalized to generate test sequences that obtain high coverage of transition delay faults with greatly reduced heat dissipation.

Several enhancements to the proposed ATPG algorithm are under investigation, such as techniques to reduce test sequence length while maintaining low heat dissipation. Also, the development of a unified ATPG, DFT design, and test application methodology for full scan circuits is being investigated. Extensions of this approach to sequential circuits are being considered as well.

Acknowledgments

This research was supported by U.S. National Science Foundation Research Initiation Award No. MIP-9210871.

References
