

# ATPG for Heat Dissipation Minimization during Scan Testing

Seongmoon Wang

Sandeep K. Gupta

E.E. – Systems, University of Southern California, Los Angeles CA 90089-2562

## Abstract

*An ATPG technique is proposed that reduces heat dissipation during testing of sequential circuits that have full-scan. The 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. The proposed ATPG exploits all don't cares that occur during scan shifting, test application, and response capture to minimize switching activity in the circuit under test. Furthermore, an ATPG that maximizes the number of state inputs that are assigned don't care values, has been developed. The proposed technique has been implemented and used to generate tests for full scan versions of ISCAS 89 benchmark circuits. These tests decrease the average number of transitions during test by 19% to 89%, when compared with those generated by a simple PODEM implementation.*

## 1 Introduction

The main objective of traditional test development has been attainment of high fault coverage. As the techniques have matured and this objective has been attained, other objectives have become important. We believe that reducing heat dissipation during test application is rapidly becoming another objective of the test development process. In this paper, we present a new *automatic test pattern generator (ATPG)* that generates tests for *full scan circuits* that minimize heat dissipation in the circuit during their application via the scan chain. Of course, the tests generated by the proposed ATPG achieve high fault coverage.

The importance of heat dissipation considerations during test development is already influencing the design of practical test methodologies. For example, it is reported in [Zor93] that one of the major considerations in test scheduling has been the fact that the heat dissipated during test application can be significantly higher (sometimes, 100-200%) than that during the circuit's normal operation.

Excessive heat dissipation occurs during test application because the correlation between consecutive test vectors is often significantly lower than that between consecutive vectors applied to a 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 memories. This is even more true for high speed systems that process digital audio and video signals, where the inputs to most modules change relatively slowly. In contrast, the correlation between consecutive test vectors generated by an ATPG is very low, since a test is generated for a given target fault without any consideration of the previous vector in the test sequence. The use of design-for-testability (DFT) techniques can further decrease the correlation between successive test vectors. Finite state machines

are often implemented in such a manner that vectors representing successive states are highly correlated [TPCD94]. However, the use of scan can significantly decrease the correlation between consecutive state vectors, since values applied to the state inputs during testing represent shifted values of test vectors and circuit responses and have no particular temporal correlation.

The fact that heat dissipation during testing can be significantly higher than that during normal operation should be viewed in conjunction with the following two trends. Firstly, to attain portability and performance, a package is selected to closely match the average heat dissipation during a circuit's normal operation [Zor]. To ensure non-destructive testing of such a circuit, heat dissipation during test must be comparable to the heat dissipated during the circuit's normal operation. Secondly, aggressive timing has made it essential to identify slow chips via delay testing. This is especially important for the growing number of circuits that are being manufactured for use in MCMs and must be tested and sold as performance certified bare die [Kee92, Par92]. Consequently, circuits are now tested at higher clock rates – if possible, at the circuit's normal clock rate (called *at-speed testing*). Hence, the heat dissipation during test application is on the rise and is fast becoming a problem that requires close attention.

In this paper, an ATPG technique is presented that reduces heat dissipation during testing of sequential circuits via full-scan. The 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. The proposed ATPG exploits all don't cares that occur during scan shifting, test application, and response capture to minimize switching activity in the CUT. Furthermore, an ATPG that maximizes the number of state inputs that are assigned don't care values, has been developed.

The tests generated by this ATPG can be used for at-speed testing of chips and bare die without running the risk of damaging the device under test by excessive heat dissipation. In case of at-speed testing of bare die for MCMs, the use of tests generated by the proposed ATPG can obviate the need for expensive heat removal equipment that may be required otherwise. Finally, during test of a bare dice, power must be supplied during the period of test through probes. The proposed tests will reduce the excessive power and ground noise caused due to the high inductance of probes by reducing the number of transitions in the circuit. This will prevent unnecessary loss of yield caused due to the limitations of probing.

Note that the heat dissipation during testing can also be decreased by incorporating additional circuitry into scan flip-flops such that their outputs hold constant values during scan shifting. However, this will increase chip area and cause performance degradation. Heat dissipation can also be decreased by dividing a long scan chain into several shorter scan chains, such that at any given time, data is shifted into only one scan chain while the contents of other chains are held constant. While this technique can reduce transitions during scan shifting without performance degradation, hardware overhead required to control multiple scan paths is appreciable. It should be noted that in contrast to above methodologies, the ATPG proposed in this paper requires no additional hardware to achieve reduction in heat dissipation.

### Design Automation Conference ®

Copyright © 1997 by the Association for Computing Machinery, Inc. Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Publications Dept, ACM Inc., fax +1 (212) 869-0481, or [permissions@acm.org](mailto:permissions@acm.org).

0-89791-847-9/97/0006/\$3.50

DAC 97 - 06/97 Anaheim, CA, USA

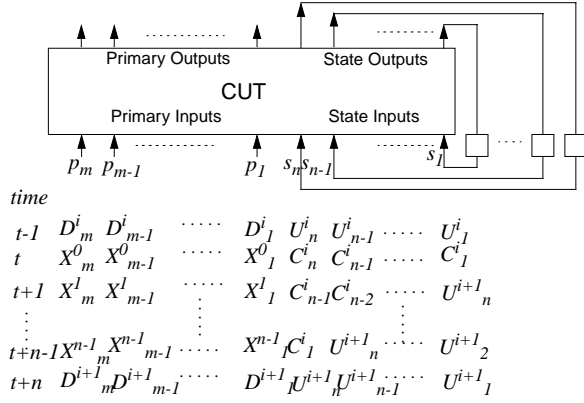


Figure 1. Application of Tests via Scan

## 2 Scan Based Testing

In this paper, we assume that the sequential circuit under test (CUT) implemented in CMOS has full-scan, and employs a single scan chain for test application. We also use the *single-stuck-at* fault model. In a such scenario, traditionally, an ATPG for combinational circuits is used to generate *combinational test vectors* by considering only the combinational part of the scan circuit under test.

Figure 1 describes the test application via scan for a CUT that has  $m$  primary inputs and  $n$  state inputs. The combinational ATPG generates a set of combinational test vectors, each of which is a binary  $m+n$  tuple and must be applied to the  $m$  primary inputs ( $p_1, p_2, \dots, p_m$ ) and the  $n$  state inputs ( $s_1, s_2, \dots, s_n$ ) during test application. The bits of a test vector that are applied to the primary and state inputs will be referred to as its *primary input* and *state input parts*, respectively. Assume that a test vector,  $V^i = D_m^i, D_{m-1}^i, \dots, D_1^i, U_n^i, U_{n-1}^i, \dots, U_1^i$ , is applied to the primary inputs and state inputs of the CUT at time  $t-1$ , and the state part of the CUT's response to  $V^i$ , say  $C_n^i, C_{n-1}^i, \dots, C_1^i$ , is captured in the scan register at time  $t$ . Subsequently, the state input part  $U_n^{i+1}, U_{n-1}^{i+1}, \dots, U_1^{i+1}$  of the next combinational vector,  $V^{i+1}$ , is shifted in during the next  $n$  cycles ( $t+1, t+2, \dots, t+n$ ) while the test response  $C_n^i, C_{n-1}^i, \dots, C_1^i$  is shifted out. This process of shifting in the state part of a test vector will be referred to as *scan shifting*. Note that no specific values need to be applied to the primary inputs at times  $t, t+1, t+2, \dots, t+n-1$ . This is depicted by vector  $X_m^j, X_{m-1}^j, \dots, X_1^j$  that is applied to primary inputs at time  $t+j$ , where  $j = 0, 1, 2, \dots, n-1$ . Finally, the primary input part of  $V^{i+1}$ ,  $D_m^{i+1}, D_{m-1}^{i+1}, \dots, D_1^{i+1}$ , is applied at  $t+n$ , and the response of the CUT to the combinational test vector  $V^{i+1}$  is captured into the flip-flops at  $t+n+1$ . This is repeated until all test vectors are applied.

The above discussion shows that the switching activity in a CUT during the application of a scan based test depends not only on correlation between the two consecutive combinational test vectors  $V^i$  and  $V^{i+1}$ , but also on how the tests are applied. Even though two vectors, such as  $V^i$  and  $V^{i+1}$ , may cause a minimum number of transitions in the CUT when applied in *two consecutive clock cycles* to the combinational part of the CUT, a significant number of transitions may occur in the circuit if they are applied to a sequential circuit via its scan chain. Hence, a sequence of test vectors that minimizes heat dissipation in a combinational circuit may not minimize heat dissipation during scan testing of the corresponding sequential circuit via its scan chain. Hence, an ATPG that considers the heat dissipated during scan shifting is required to generate test sequences that reduce heat dissipation during scan testing of sequential circuits.

A combinational test vector, e.g.,  $V^{i+1}$ , generated by an ATPG

for a target fault, is not fully specified and the target fault can be detected independent of the binary value assigned to each of the unspecified inputs. In addition to the don't cares in combinational vector  $V^{i+1}$ , the  $n$   $m$ -tuples depicted as  $X_m^j, X_{m-1}^j, \dots, X_1^j$  ( $j = 0, 1, \dots, n-1$ ) in Figure 1, applied to the primary inputs during scan shifting, are not specified by the test vector  $V^{i+1}$ . This implies that all primary inputs during scan shifting can be treated as *don't cares* as well.

Don't cares in the state input part have different characteristics from those at primary inputs during scan shifting. Primary input don't cares are fully controllable, i.e. completely independent binary  $m$  tuples can be assigned to these don't cares during each scan shift cycle. On the other hand, only one binary  $n$  tuple is shifted into the scan register for each test vector.

The proposed ATPG assigns each primary input a value that implies the *controlling value* at the inputs of the gates that are fed by that primary input as well as state input(s) to *block* the transitions caused by shifting of the scan chain contents. (The controlling value of a gate is the binary value which, when applied to any input of a gate, determines the output value of that gate *independent of the values applied to the other inputs of the gate*.) In the following, first the considerations and a procedure for the assignment of don't cares to the primary inputs will be discussed. Subsequently, an ATPG procedure that generates tests with minimum number of specified state inputs will be described, followed by a discussion of the overall test generation strategy.

## 3 Primary Inputs during Scan Shifting

Consider a primary input  $p_j$  of a circuit. Associated with input  $p_j$  are the don't care values,  $X_j^0, X_j^1, \dots, X_j^{n-1}$ , during scan shifting, i.e. at times  $t, t+1, \dots, t+n-1$ . One possible strategy is to determine, for each  $p_j$ , a *single* binary value and apply it to that input for *each scan shifting clock of each vector*. This strategy has two main advantages over a strategy that assigns new values to the don't cares at primary inputs for each test vector. Firstly, in the former strategy, an appropriate binary assignment can be determined once for a given circuit and used for each combinational vector, reducing the run time complexity of the overall ATPG. More importantly, such binary values can be *implicitly stored* in an intelligent automatic test equipment, thereby drastically decreasing the test data volume. In the following, it will first be shown that a fixed value can be assigned to a large number of primary inputs in a manner that guarantees the minimization of the number of transitions, independent of the specific sequence of combinational vectors generated by ATPG. Techniques to identify such inputs and an appropriate binary assignment are presented. A technique that takes into account the specific test vector being applied to assign binary values to the remaining primary inputs is then presented.

### 3.1 Notations and Definitions

#### 3.1.1 Basic Circuit Definitions

If a line  $l$  of a circuit  $C$  is driven by a gate  $g$ , then  $l$  is said to be a *fanout* of  $g$ . The *transitive fanout* of a line  $l$  includes all lines reachable from  $l$  via forward traversal of a sequence of gates and fanouts. A *path* is a sequence of consecutive circuit lines. The *inversion parity* of a path is the number of inverting gates along the path, modulo-2. If the inversion parity of a path is 1, then the path is said to have *odd inversion parity*, otherwise the path is said to have *even inversion parity* [ABF90]. The *forward cone* of line  $l$  is the set of all gates and circuit lines in the transitive fanout of  $l$ . During test generation, each line  $l$  in a circuit is assigned one of three values (0, 1,  $X$ ). (Initially, all lines are assigned  $X$ .) A path along which all lines have unknown values ( $X$ ) is called an *x-path*. The assignment of a binary value at  $l_1$  can imply a desired value at  $l_2$  only if there exists at least one  $x$ -path from  $l_1$  to  $l_2$ .

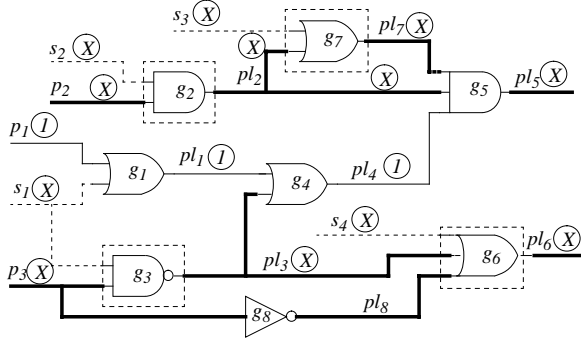


Figure 2. Example Circuit  $C$

**Definition 1**  $C$  = combinational part of a sequential CUT with binary values assigned to some or none of its lines

$XPI = \{p_i \mid p_i \text{ is a primary input of } C \text{ that is not assigned a binary value}\}$

$SI = \{s_i \mid s_i \text{ is a state input of } C\}$

$TPI = \{pl \mid pl \text{ is a line in the transitive fanout of any } p_i \in XPI\}$

$TPI^C = \{pl \mid pl \text{ is a line not in the transitive fanout of any primary input}\}$

$TPI' = \{pl \mid pl \in TPI \text{ and } \exists \text{ an x-path from any } p \in XPI \text{ to } pl\}$

□

### 3.1.2 Blocking Objectives

If none of the inputs of a gate is assigned the controlling value of the gate, then the gate is called an *unblocked gate*. If a transition propagates to the output of a gate  $g$ , then new transitions are caused at each fanout of  $g$ . Furthermore, transitions at the fanouts may cause transitions at the output of the gates driven by them, and so on. The transition caused by a state input should hence be blocked as close to the state input as possible, to prevent it from propagating into the forward cone of the state input. Let  $G$  be the set of gates that have at least one input in  $TPI'$  and at least one input in  $TPI^C$ ; the elements of  $G$  are called *blocking objectives*. The set  $G_{p_i}$  contains the blocking objectives that are in the forward cone of primary input  $p_i$ ; gates in  $G_{p_i}$  are called *blocking objectives with respect to*  $p_i$ .

**Definition 2**  $C$  = combinational part of a sequential CUT with binary values assigned to some or none of its lines

$UB = \{ub \mid ub \text{ is a gate with none of its inputs assigned the gate's controlling value}\}$

$G = \{g \mid g \in UB \text{ with at least one input in } TPI' \text{ and at least one in } TPI^C\}$

$G_{p_i} = \{g \mid g \in G \text{ and } \exists \text{ an x-path from } p_i \text{ to at least one input of } g\}, \forall p_i \in XPI$

□

**Example 1** Figure 2 shows the combinational part of a sequential circuit that has primary inputs  $p_1, p_2$ , and  $p_3$  and state inputs  $s_1, s_2, s_3$ , and  $s_4$ .  $SI = \{s_1, s_2, s_3, s_4\}$ . Suppose that primary input  $p_1$  is assigned a 1 to block gate  $g_1$ . Circled 1 or  $X$  at a line denotes the current value assigned to the line. With  $p_1$  assigned a 1,  $XPI = \{p_2, p_3\}$ . The lines in  $TPI$ ,  $p_2, p_3, pl_3, pl_5, pl_6, pl_7, pl_8$ , are denoted by thick lines and the lines in  $TPI^C$ ,  $s_1, s_2, s_3, s_4$ , are denoted by dotted lines. All elements of  $TPI$ , except  $pl_4$ , which is assigned a binary value (no x-path exists from any primary input to  $pl_4$ ), are also members of  $TPI'$ .

□

### 3.2 Independent Inputs

If primary input  $p_i$  can not be used to block transitions caused by a scan inputs during scan shifting (i.e. if  $G_{p_i} = \emptyset$ ) and any transition caused by a scan input does not propagate to any gates in the forward cone of  $p_i$ , then the primary input  $p_i$  is called *independent*. Independent primary inputs are assigned binary values that minimize transitions based on the binary value assigned to the in-

puts in the preceding test vectors. Appropriate values for independent inputs are determined during each test clock (e.g. clock  $t - 1$  in Figure 1) by using the *don't care assignment* algorithm described in [WG94]. The determined values are maintained until the application of next test.

### 3.3 Single Input Conflict Free Assignment

If a binary assignment at  $p_i$  can be made such that it helps block all gates in  $G_{p_i}$ , the assignment at  $p_i$  is called *conflict free*. Let  $g_a$  be any gate that belongs to  $G_{p_i}$ . Let  $\{xp_1, xp_2, \dots, xp_h\}$  be the set of x-paths from  $p_i$  to the inputs of  $g_a$ . Let  $\Pi_{a,i}$  be the parity of the x-path  $xp_i$ ,  $i = 1, 2, \dots, h$ . Finally let  $c_a$  be the controlling value of  $g_a$ .

**Lemma 1** The assignment of primary input  $p_i$  is *conflict free* if and only if  $\Pi_{a,i} \oplus c_a$  are identical,  $\forall i = 1, 2, \dots, h$  and  $\forall g_a \in G_{p_i}$ . □

**Example 2** In Figure 2, the assignment at  $p_3$  is conflict free if it helps block all gates in  $G_{p_3}$ , i.e.  $g_3$  and  $g_6$ . X-paths ( $p_3, g_3, pl_3$  and  $p_3, g_3, g_8, pl_8$ ) from  $p_3$  to  $pl_3$  and  $pl_8$  have odd parity and the controlling value of  $g_6$  is 1. Hence, setting  $p_3$  to 0 will help block transitions at the outputs of  $g_6$  as well as  $g_3$ . In other words, assigning 0 to  $p_3$  is good in terms of blocking all gates in  $G_{p_3}$ . The assignment of 0 to  $p_3$  is hence conflict free.

Now consider a modified circuit obtained by replacing the NAND gate  $g_3$  in  $C$  with an AND gate. The parity of the path  $p_3, g_3$ , and  $pl_3$  now becomes even. Since the path parity is even and a 1 must be assigned  $pl_3$  to block  $g_6$ , a 1 is required at  $p_3$  when backtraced along the above x-path. This is in conflict with the value 0 that is required to block  $g_3$ . Hence, in the modified circuit,  $p_3$  can not be assigned in a conflict free manner. □

The above definition of conflict free assignment can be expanded in three ways. Firstly, assigning a binary value to a primary input may set lines on one or more x-paths to binary values, consequently removing some lines from  $TPI'$  and some gates from  $G$ . Consequently, primary inputs that were previously not conflict free can be assigned in a conflict free manner after some primary input assignments. Secondly, since all scan inputs  $s_i \in SI$  are assumed to be *uncontrollable* [AKR91] during the assignment of primary input don't cares (due to the fact that the values of the scan inputs are not known during this analysis, which is performed *before* the tests are generated), there may exist gates in  $G$  none of whose inputs can be set to the gates' controlling values by assigning any combination of binary values to the primary inputs. Since such gates are *unblockable*, they can be removed from  $G_{p_i}$  to make additional conflict free assignments possible [WG96]. Finally, the uncontrollability of state inputs can render some x-paths from primary input  $p_i$  to the gates in its blocking objective without any *influence*. Elimination of such x-paths can also make additional conflict free assignment possible [WG96].

### 3.4 Multiple Input Conflict Free Assignment

Even when no more conflict free primary input assignments can be found by considering each primary input individually, additional conflict free primary input assignments can be found by considering simultaneously multiple primary inputs. We have developed efficient techniques to make conflict free multiple input assignments [WG96].

### 3.5 Iterative Improvement Heuristic

The primary inputs that are not identified as independent, or assigned conflict free (single/multiple) values, are assigned binary values to maximize blocking. This is achieved by using Kernighan and Lin [KL70] iterative improvement bipartitioning algorithm (*K-L algorithm*). Associated with each gate  $g_a$  whose output is currently assigned  $X$  is a weight  $w(g_a)$ , the number of lines in the forward cone of  $g_a$  that are not yet assigned binary values. The objective of

this algorithm is the maximization of the following function:

$$F(X_j) = \sum_{\forall g_a} B(g_a) \times w(g_a), \quad (1)$$

where  $X_j$  is the vector applied to the primary inputs and  $B(g_a)$  is a function that evaluates to 1 if  $g_i$  is blocked, otherwise it evaluates to 0. Primary inputs are divided into two partitions,  $\Psi_0$  and  $\Psi_1$ , such that  $\Psi_0$  ( $\Psi_1$ ) contains primary inputs that are assigned 1(0). The primary inputs that are already assigned binary values by conflict free assignments are assigned to  $\Psi_0$  or  $\Psi_1$ . Initially, the other primary inputs are placed into  $\Psi_0$  ( $\Psi_1$ ) arbitrarily. In each iteration, a primary input  $p_i$ , that was not assigned a binary value by conflict free assignment, is moved from  $\Psi_v$  to  $\Psi_{\bar{v}}$  ( $v = 0, 1$ ). In other words, the bit corresponding to  $p_i$  in the initial vector  $X_{init}$  is flipped (denoted by  $X_{p_i \leftarrow \bar{p}_i}$ ) and the gain ( $\Delta F(X_{p_i \leftarrow \bar{p}_i})$ ) due to flipping is calculated, where  $\Delta F(X_{p_i \leftarrow \bar{p}_i})$  is given by:

$$\Delta F(X_{p_i \leftarrow \bar{p}_i}) = F(X_{init}) - F(X_{p_i \leftarrow \bar{p}_i}). \quad (2)$$

The migration is repeated until  $F(X_{p_i \leftarrow \bar{p}_i})$  does not increase any longer.

#### 4 Scan Input Assignment to Minimize the Number of Transitions

The contents of the scan register during scan shifting are determined by the values captured in response to the test vector applied and the new test being scanned in. The next test generated may have don't cares. The switching activity in the CUT during test application can be reduced by carefully assigning these don't cares. Hence, combinational test vectors with the maximum number of don't cares in their state input part are more suitable for minimizing transitions in circuit lines. An existing implementation of PODEM [Goe81] has been modified to generate combinational test vectors that have minimum number of specified bits at the state inputs. New cost functions: controllability and observability, are defined to direct the ATPG to generate such tests. These cost functions are calculated only once during the entire test generation process in a preprocessing step.

The controllability cost  $Cv(l)$  is the minimal number of state inputs that need to be assigned to set the line  $l$  to a desired value  $v$ . In order to detect the stuck-at- $v$  fault at line  $l$ , first the target fault is activated by setting  $l$  to  $\bar{v}$ . Subsequently the activated fault effect is propagated to a primary or state output. Therefore, generating a test vector consists of many *line-justifications*. In PODEM [Goe81], a value  $v$  at a line  $l$  is justified by mapping  $v$  to input (primary and state) assignments by backtracing to inputs. Whenever there is a choice of several paths to backtrace from a target line to the inputs, the controllability cost functions are used by the ATPG to select backtrace paths that require minimum number of state input assignments. The controllability cost is given by

$$Cv(l) = \begin{cases} 0, & \text{if } l \text{ is a primary input} \\ 1, & \text{if } l \text{ is a state input} \\ |\bigcup_{l_j} \{s \mid s \text{ are min state to set inputs} \\ \text{required to set } l_j \text{ to } \bar{c}_a\}|, & \text{if } v = \bar{c}_a \oplus i_a \\ \min_j \{Cv(l_j)\}, & \text{if } v = c_a \oplus i_a \end{cases} \quad (3)$$

where  $l_j$  are the inputs of the gate  $g_a$  with output line  $l$ , and  $c_a$  and  $i_a$  are the controlling value and inversion of  $g_a$ , respectively. To take into account the differences due to the order in which flip-flops appear in the scan chain, the controllability cost function can be refined by assigning suitable weights to each state input [WG96].

During test generation, a gate whose output value is currently unknown and at least one of whose inputs has the fault effect belongs to *D-frontier* [Goe81]. In the proposed ATPG, a gate that is

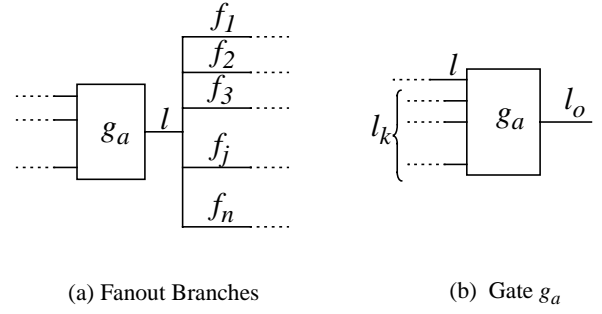


Figure 3. Gate Model

likely to need the least number of state input assignments to propagate the fault effect to its output is selected from D-frontier repeatedly, until the fault effect reaches one or more primary or scan outputs. A gate in the D-frontier whose input with fault effect has minimum observability cost is selected each time. The observability cost function  $O(l)$  indicates the minimum number of state inputs that need to be assigned binary values to propagate a value at line  $l$  to an observation point. Observability is calculated for every line in the CUT, starting from primary and state outputs and traversing the circuit backward toward primary and state inputs and is given by

$$O(l) = \begin{cases} 0, & \text{if } l \text{ is an observation point} \\ \min_j \{O(f_j)\}, & \text{if } l \text{ is a fanout stem} \\ |\bigcup_{l_k \neq l} \{s \mid s \text{ are min state inputs} \\ \text{required to set } l_k \text{ to } \bar{c}_a\}| + O(l_o), & \text{otherwise} \end{cases} \quad (4)$$

where  $f_j$  are fanout branches of line  $l$  and  $l_k$  are inputs of gate  $g_a$  that is driven by  $l$ , and  $l_o$  is the output of  $g_a$  (Figure 3).

The objective of the proposed ATPG is to generate a test vector that has the maximum number of unspecified state inputs. As described earlier, in order to generate a vector that detects the *stuck-at-v* fault at line  $l$ , the fault should be activated by setting  $l$  to  $v$ , then the fault effect should be propagated to one or more outputs. Controllability and observability correspond to the former and the latter, respectively. Thus a test vector that specifies boolean values at a minimum number of state inputs is obtained in *fanout free circuits* by using the two proposed cost functions for (a) selection of objective, and (b) directing the backtrace procedures in PODEM [Goe81]. However, in circuits with fanouts, a test vector that is generated by the proposed ATPG may not have minimum number of specified states.

In general, a test vector generated by the proposed ATPG has many don't cares at state inputs which can be assigned to minimize the switching activity in the circuit. Again this can be performed by using the K-L algorithm [KL70]. This procedure is similar to that used to find the primary input pattern that can block the most gates (Section 3.5), except that the objective function and the gain are defined somewhat differently.

Since the time complexity of the K-L algorithm is mainly determined by the number of state inputs, the run time for circuits that have many state inputs (say, greater than 50) may not be acceptable. A simple heuristic can be used for these circuits instead of the K-L algorithm without increasing the number of transitions significantly. Assume that  $n$  consecutive scan flip-flops in a scan chain are assigned don't cares in a test vector and are flanked by two flip-flops  $s_i$  and  $s_j$  that are assigned the same binary value  $v$  in the test vector. The simple heuristic assigns  $v$  to all these don't cares. If  $s_i$  and  $s_j$  are assigned different values,  $v$  and  $\bar{v}$ , in the test vector, then the simple heuristic chooses a value randomly and assigns it to these don't cares. Experimental results with these two state input don't care assignment procedures show that tests generated by the proposed ATPG which uses the simple heuristic cause, on an average, the same number of transitions as those generated by the proposed

ATPG which uses the K-L algorithm. While the run time is lower for the simple heuristic, the length of the test sequence obtained is typically higher. This is due to the fact that the simple heuristic tries to assign the same binary values to adjacent scan flip-flops. Many scan inputs are hence assigned the same binary value for most test vectors resulting in lower fault coverage. Hence, to achieve the same overall fault coverage, longer test sequences are required. Enhancements to reduce test sequence length are under investigation [WG96].

Tests for some faults need specific values at many state inputs. Though don't cares in such tests are not enough to reduce transitions, they can be used to detect additional faults. If the number of specified state inputs in a test generated using the abovementioned ATPG is greater than a predefined number (e.g. 80%), the generated test is discarded and the fault is moved to a *high cost fault list*, that is initially empty. Target faults are taken from the *regular fault list* until the regular fault list is empty. After the regular fault list is empty, target faults are selected from the high cost fault list. In order to detect additional faults by specifying any don't cares, the ATPG selects secondary fault faults and generates a test for these secondary faults until all don't care are specified. This procedure is repeated until the ATPG tries all faults in the high cost fault set or all the don't cares are assigned binary values.

The proposed test generation algorithm can now be described as follows.

1. Perform uncontrollability analysis (Section 3.3).
2. Assign appropriate binary values to primary input don't cares that can be assigned in a conflict free manner, either one at a time or as a group (Sections 3.3 and 3.4). These assignments are used for all vectors and during all scan shifting clocks.
3. Identify all independent primary inputs (Section 3.2). and assign binary values to these inputs once, for each vector.
4. Apply the bipartitioning algorithm to find an assignment for the remaining primary input don't cares during scan shifting to block most gates in the CUT (Section 3.5).
5.
  - a) If the regular fault list is empty, go to Step 6.
  - b) Select a target fault from the regular fault list and generate a combinational test  $V^{i+1}$  using the proposed ATPG that assigns minimum number of state inputs. If the generated test has fewer don't cares in state input part than a predefined number, then move the target fault to the high cost fault list and go to Step 5 a).
  - c) Assign remaining unspecified primary inputs (which are independent primary inputs identified in Step 3) according to their previous values using the procedure outlined in [WG94]. (The values assigned at this time are held constant during scan shifting of vector  $V^{i+1}$ .)
  - d) Apply either the simple heuristic or the K-L algorithm to assign binary values to the remaining state input don't cares to minimize the number of transitions.
  - e) Perform fault simulation and drop detected faults from the fault list and go to Step 5 a).
6.
  - a) If there are no more undetected faults in the high cost fault list, then exit. Otherwise, select a target fault from the high cost fault list.
  - b) Generate a combinational test for the selected target fault by using a normal PODEM.
  - c) If the generated test has any don't cares in its state input part, select a secondary target fault from the high cost fault list and generate a test for it by assigning binary value to don't cares. Repeat this step until there are no remaining don't cares in the state input part or all faults in the high cost fault list are tried.
  - d) Go to Step 6 a).

**Table 1. Experimental Results**

CKT	FC		Avr. # Trans.		# Vect.		Eff.	Time (sec)	
	N	P	N	P	N	P	Red.	N	P
s208	100	100	40.1	14.3(64)	38	55	.51	.2	.5
s298	100	100	105.4	46.0(56)	47	68	.63	.3	.6
s344	100	100	109.5	50.0(54)	30	72	1.1	.3	.7
s386	100	100	112.5	83.0(26)	82	86	.77	.7	2.4
s420	100	100	68.0	14.2(79)	65	114	.37	.6	1.7
s444	99.1	99.1	135.5	43.6(67)	38	92	.78	.5	1.2
s510	100	100	141.1	114.4(19)	69	80	.94	.9	2.9
s526	100	100	178.4	74.9(58)	78	142	.76	1.1	2.4
s641	100	99.7	206.6	22.9(89)	83	136	.21	3.1	2.7
s713	97.5	97.0	221.4	24.2(89)	69	118	.19	11.4	13.2
s820	100	100	283.0	184.5(35)	146	192	.86	3.8	12.2
s832	99.8	99.5	286.9	185.5(35)	156	184	.76	4.4	12.2
s838	100	100	106.5	13.6(87)	131	234	.22	3.2	9.6
s953	100	100	205.7	27.8(86)	114	161	.19	4.1	6.6
s1196	100	99.5	346.8	47.0(86)	180	219	.16	8.6	12.3
s1423	98.6	98.1	477.0	122.2(74)	85	237	.71	10.1	14.9
s1488	100	100	500.9	393.8(21)	164	188	.90	11.4	35.1
s1494	99.9	99.5	506.4	403.9(21)	159	185	.93	11.5	38.1
s5378	99.3	99.0	1545.8	321.5(79)	315	739	.49	158	401
s9234	93.1	93.1	2785.0	716.1(74)	351	936	.68	613	1467

## 5 Experimental Results

Table 1 shows the experimental results for full scan versions of ISCAS89 sequential benchmark circuits. The experiments were performed on a Sparcstation 4 with 32Mbytes of memory. Table 1 compares the results obtained by the normal implementation of PODEM (columns labeled *N*) and those obtained by the proposed ATPG (columns labeled *P*). Fault coverages (heading *FC*), average number of transitions (heading *Avr. # Trans.*), total number of test vectors in test sequences (heading *# Vect.*), and test generation times (heading *Time*) are presented. The number of transitions was counted under zero delay model. Zero delay model is used exclusively by test generators and fault simulators for stuck-at faults. Under the zero delay model, no hazards are considered at the circuit lines and the heat dissipation is assumed to be mainly due to  $0 \rightarrow 1$  and  $1 \rightarrow 0$  transitions at the circuit lines. The use of zero delay model is justified by the observation that the heat dissipation estimated under this model has a high correlation with that under the general delay model [SGDK92]. Column labeled *Eff. Red.* (Effective Reduction) shows the total number of transitions in each CUT during entire scan based test when the test vectors generated by the proposed ATPG are applied, as a fraction of those occurring due to the application of the tests generated by the normal PODEM. The simple heuristic was used to assign binary values to unspecified state inputs. In the normal PODEM, all don't cares were assigned randomly. First note that the fault coverage obtained by using two ATPGs are *almost identical*.

The results demonstrate that the tests generated using the proposed ATPG decreased the average number of transitions by 19% to 89%. The numbers shown in parenthesis (under the heading *Avr. # Trans.*, and column *P*) denote the percentage increase/decrease over the results obtained by the normal PODEM. These results clearly show that the switching activity during test application can be reduced significantly by using the proposed ATPG. It should be noted that larger reductions occur in circuits that have more primary and state inputs (s641, s713, s838, and s913). This is due to the fact that in such circuits, more primary inputs can be assigned binary values to block more gates that may otherwise have transitions. Furthermore, for such circuits, most of test vectors generated by the proposed ATPG have many don't cares in state input part. If these don't cares are carefully assigned to minimize transitions, only a small portion of state inputs will have transitions during scan shifting resulting in lower average number of transitions. The number of transitions also depends on the circuit structure. For example, s420 and

s510 have the same number of primary inputs and state inputs, but the results are significantly different. This is due to the difference in the structures of these circuits. For example, 14 primary input can be assigned in a conflict free manner for s420 as opposed to 5 for s510.

The number of vectors in test sequences increased by a factor of 1 to 2.8. To take into account both the reduction in average number of transitions and the increase in test length, we present the effective reduction in the Table. This shows that the reduction in the average number of transitions are more than sufficient to compensate for the increase in the test sequence for all circuits except s344. For s344, only two primary input don't cares can be assigned a binary value to block transitions at state inputs and all other primary inputs are independent of state inputs. The results for these circuits can be improved at the cost of longer run time by using the K-L algorithm to assign state input don't cares. The effective reductions for s344, when the K-L algorithm is used instead of the simple heuristic, is 0.56 while the run time is 43.8 seconds. The K-L algorithm outperformed significantly the simple heuristic in the circuits (such as s344) which have small number of state inputs and very few of whose primary inputs can be used to block transitions at state inputs. For these circuits, the run time of the proposed ATPG using the K-L algorithm is within a few tens of seconds higher than that of the simple heuristic.

Finally, the run time of the proposed ATPG (with the simple heuristic) is a factor of about 2 to 3 higher than that of the normal PODEM, for the most circuits. Considering that the test sequence length also increases in a similar proportion, these data clearly demonstrate that the time complexity of the operation to reduce transitions in the CUT is very low.

## 6 Conclusion

An ATPG technique is proposed that reduces heat dissipation during testing of sequential circuits that have full-scan. The 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. The proposed ATPG exploits all don't cares that occur during scan shifting, test application, and response capture to minimize switching activity in the CUT. Furthermore, an ATPG that maximizes the number of state inputs that are assigned don't care values, has been developed. In order to guide the ATPG to generate test vectors that have maximum number of don't cares at state inputs, new controllability and observability cost functions have been defined and used to guide backtrace and to select objectives from D-frontier. Don't cares at primary inputs during scan shifting and capture are used to block gates that may cause transition during scan shifting and don't cares at state inputs are assigned binary values that cause the minimum number of transitions.

The proposed algorithm has been implemented and the generated tests are compared with those generated by a simple PODEM implementation for full scan versions of ISCAS 89 benchmark circuits. Tests generated by the proposed ATPG decreased the average number of transitions during test by 19% to 89%. with higher reductions occurring in circuits that have more primary and state inputs. Since large circuits typically have many primary and state inputs, higher reduction in the number of inputs will be obtained in large circuits (79% and 74% for s5378 and s9234). Even though the large circuits have very small number of primary inputs, compared with the number of state inputs, since the number of state inputs is large, the proposed ATPG can still generate tests which significantly reduce the number of transitions. This is due to the fact that in such a circuit, only a few of state inputs need to be specified for most faults and the don't cares can be exploited to reduce transitions. These reductions in circuit transitions were achieved with a reasonable increase in the time complexity of the ATPG (factor of 2 to 3). We believe that the increase in run time is mainly due to the longer test

sequence length and are investigating enhancements to reduce the test length.

It should be noted that with little modification, the proposed ATPG can be used for stress test (opposite application) which requires test vectors which maximize the number of transitions during test. The proposed ATPG has been modified to obtain this objective and experimental results show that the number of transitions can also be increased significantly.

## REFERENCES

- [ABF90] M. Abramovici, M. A. Breuer, and A. D. Friedman. *Digital Systems Testing and Testable Design*. Computer Science Press, New York, N.Y., 1990.
- [AKR91] M. Abramovici, J. J. Kulikowski, and R. K. Roy. The Best Flip-Flops to Scan. In *Proceedings IEEE International Test Conference*, pages 166–173, October 1991.
- [Goe81] P. Goel. An Implicit Enumeration Algorithm to Generate Tests for Combinational Logic Circuits. *IEEE Transactions on Computers*, Vol. C-30(3), March 1981.
- [Kee92] D. C. Keezer. Bare Die Testing and MCM Probing Techniques. In *Proceedings Multi Chip Module Conference*, pages 20–23, March 1992.
- [KL70] B. W. Kernighan and S. Lin. An Efficient Heuristic Procedure for Partitioning Graphs. *Bell System Technical Journal*, 49:291–307, February 1970.
- [Par92] R. H. Parker. Bare Die Test. In *Proceedings Multi Chip Module Conference*, pages 24–27, March 1992.
- [SGDK92] A. Shen, A. Ghosh, S. Devadas, and K. Keutzer. On Average Power Dissipation and Random Pattern Testability of CMOS Combinational Logic Networks. In *Proceedings IEEE International Conference on Computer-Aided Design*, pages 402–407, November 1992.
- [TPCD94] C.-Y. Tsui, M. Pedram, C.-A. Chen, and A. M. Despain. Low Power State Assignment Targeting Two- and Multi-level Logic Implementation. In *Proceedings IEEE International Conference on Computer-Aided Design*, pages 82–87, 1994.
- [WG94] S. Wang and S. K. Gupta. ATPG for Heat Dissipation Minimization During Test Application. In *Proceedings IEEE International Test Conference*, pages 250–258, October 1994.
- [WG96] S. Wang and S. K. Gupta. ATPG for Heat Dissipation for Scan Testing. *University of Southern California Computer Engineering Technical Report 96-22*, 1996.
- [Zor] Y. Zorian. Private Communication.
- [Zor93] Y. Zorian. A Distributed BIST Control Scheme for Complex VLSI Devices. In *Proceedings VLSI Testing Symposium*, pages 4–9, 1993.