

8-1-1994

Automatic Test Reordering for Electrical and Thermal Stress during Monitored Burn-in

Kaushik Roy

Purdue University School of Electrical Engineering

Rabindra K. Roy

Purdue University School of Electrical Engineering

Abhijit Chatterjee

Purdue University School of Electrical Engineering

Follow this and additional works at: <http://docs.lib.purdue.edu/ecetr>

Roy, Kaushik; Roy, Rabindra K.; and Chatterjee, Abhijit, "Automatic Test Reordering for Electrical and Thermal Stress during Monitored Burn-in" (1994). *ECE Technical Reports*. Paper 196.

<http://docs.lib.purdue.edu/ecetr/196>

This document has been made available through Purdue e-Pubs, a service of the Purdue University Libraries. Please contact epubs@purdue.edu for additional information.

AUTOMATIC TEST REORDERING FOR
ELECTRICAL AND THERMAL STRESS
DURING MONITORED BURN-IN

KAUSHIK ROY
RABINDRA K. ROY
ABHIJIT CHATTERJEE

TR-EE 94-27
AUGUST 1994



SCHOOL OF ELECTRICAL ENGINEERING
PURDUE UNIVERSITY
WEST LAFAYETTE, INDIANA 47907-1285

Automatic Test Reordering for Electrical and Thermal Stress during Monitored Burn-in

Kaushik Roy, *Member IEEE*

School of Electrical Engineering
Purdue University
West Lafayette, IN 47907-1285, USA

Rabindra K. Roy, *Member IEEE*

C&C Research Laboratories
NEC USA
Princeton, NJ 08540-6620, USA

Abhijit Chatterjee, *Senior Member IEEE*

School of Electrical Engineering
Georgia Institute of Technology
Atlanta, GA 30332-0250, USA

Key Words — Burn-in, Reliability, Testing, Circuit Activity, power dissipation

Reader Aids —

General purpose: Present methods for efficient Burn-in of VLSI chips

Special math needed to understand explanations: Boolean algebra and graph theory

Special math needed to use results: Same

Results useful to: IC reliability and test engineers

Abstract

As the complexity of VLSI circuits increases, the semiconductor manufacturers progress towards in-situ monitored burn-in to improve the quality and reliability of their products. A new method is proposed to control the switching current drawn by a CMOS circuit during burn-in and test applications. This is based on reordering of the test vectors such that the circuit activity or electrical stress is modified as specified by the designer. or the quality control engineer during monitored burn-in. Results on several ISCAS-89 benchmarks show that the current (or power dissipation) requirement of a circuit can be changed by more than a factor of 4 by reordering the input test vectors. We have developed a CAD tool which can reorder tests for monitored burn-in to achieve average switching current within 5% of

the specified value. The technique can also be used to selectively produce higher switching activity in specified portions of the circuit to produce sharp temperature gradient inside the chip. Such temperature gradient can weed out defects such as crystal anomalies, junction imperfections etc, which might otherwise show up during the infant mortality period.

1 Introduction

Due to the rapid development of VLSI, there has been a major thrust toward improving the quality and reliability of VLSI circuits. Even the best design and fabrication techniques can propagate some reliability and/or device quality problems with the end product. In order to ensure product quality, testing is done at various levels during the design process and circuits failing the test are discarded. It has been observed that after the chip is fabricated, the failure rate follows the *bath tub* curve as shown in Figure 1. During the early life region or the *infant mortality period* the failure rate is much higher. Burn-in has been found to be the most effective screen in weeding out infant mortalities [1, 2, 3]. It simulates worst case operations of the devices accelerated through a time, power, and temperature relationship. Several different types of burn-ins have been used in the industry: *static*, *dynamic*, and *monitored*. Static burn-in is most effective in weeding out devices with thermally activated surface related defects. In dynamic burn-in, all signal lines are continually sequenced, resulting in higher power dissipation, current densities, and chip temperature than static burn-in. It has been observed in [3] that larger current densities can stress defects such as epitaxial and crystal imperfections, metallization, oxide, and junction anomalies. Many of these defects require localized thermal stresses to provide activation for the associated failure mechanism. Due to long test times in case of large circuits, testing during dynamic burn-in is becoming increasingly widespread. Such a technique is called monitored burn-in. In this paper we consider controlled power dissipation in different sections of the circuit under test during monitored burn-in. With such a technique it is possible to have varying current stress in different parts of the circuit. Such current stress can detect many latent soft defects.

For functional verification of a circuit, test vectors are generated targeting a particular

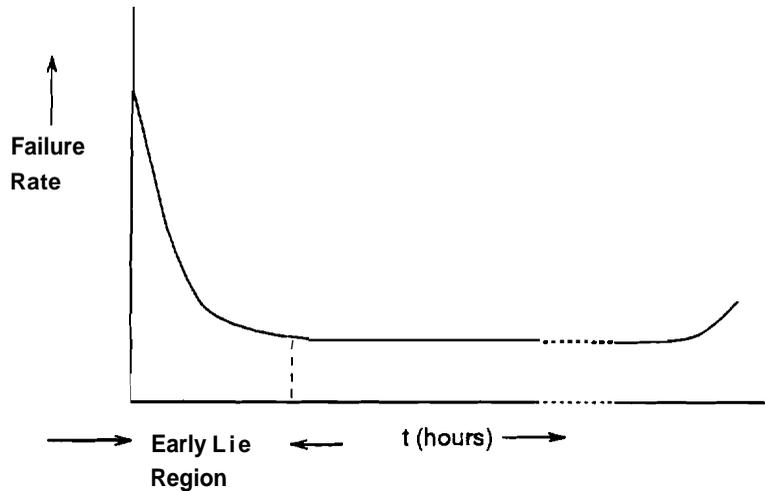


Figure 1: Bathtub curve for failure rate vs. time

fault model such as the stuck-at faults. The test vectors should be such that 100% fault coverage is achieved for the list of faults under consideration. For monitored burn-in, the test vectors can be reordered so as to generate the required activity in sections of a CMOS circuit. We have developed a graph based test reordering technique which can reorder test vectors for burn-in applications.

Recently, tremendous research effort is spent in estimating and minimizing power dissipation for portable electronic and communications applications [6, 7, 8]. However, our research effort is directed toward achieving controlled power dissipation during test applications. A graph theoretic approach to stuck-at test vector reordering will be used to obtain the optimum power dissipation. The methodology is also able to produce different current stresses in different parts of the circuit under test. Recently Chou et. al. [5] considered scheduling of tests so as to minimize power dissipation. They provide theoretical results for the power constrained test scheduling problem. The objective was to minimize total test length subject to the power constraint.

The paper is organized as follows. Section 2 describes the power dissipation model used in this paper. Section 3 considers the graph based formulation of the problem to achieve controlled power dissipation during test applications. The activity simulator which is used in estimation of power dissipation is described in Section 4. Section 5 presents the results of

our work on several ISCAS-89 benchmark examples. Finally, the results of our analysis are given in Section 6.

2 Preliminaries

As noted in Section 1, it is important to be able to control the power dissipation of a circuit or portions of it during monitored burn-in to weed out certain defects, thereby increasing the reliability of the integrated circuit. Larger power dissipation in a CMOS circuit is associated with higher switching current. Such switching current is produced by logic transitions (or *activity*) at internal nodes of the logic circuit. Activity is also a measure of the *stress* that the circuit is under. Iyer *et. al.* [4] have shown that higher stress can induce higher failure rate in a circuit. It should also be observed that higher stress corresponds to higher power dissipation and switching current, which in turn increases the temperature of the chip. In this section, we briefly describe power dissipation in CMOS circuits.

Power Dissipation in CMOS: The three sources of power dissipation in CMOS circuits are due to leakage current, short-circuit current, and switching current associated with charging or discharging of load capacitances. The latter component accounts for majority of the power dissipation in CMOS and will only be considered in our discussions. The capacitances internal to a logic gate are assumed to be small and are neglected in our analysis. The average power dissipation in a multi-level logic circuit is given by

$$P_{avg} = \frac{1}{2} V_{dd}^2 \sum_{i=1}^n A_i C_i \quad (1)$$

where V_{dd} is the supply voltage and is constant, C_i is the capacitance associated with node i of the circuit, and A_i is the activity of node and is equal to the average number of transitions per unit time. The summation is taken over all the n nodes of the circuit. From Equation 1 it is clear that the average current I_{dd} drawn from the supply voltage due to the switching component of power is equal to $I_{dd} = \frac{1}{2} V_{dd} \sum_{i=1}^n A_i C_i$. Hence, both average power dissipation and switching current are proportional to the weighted switching activity given by $\sum_{i=1}^n A_i C_i$.

For a gate level description of the circuit, the load capacitance of each gate is approximated by the fanout times the transistor input capacitance. However, estimation of signal activity at the nodes is not trivial. All published methods of estimation of signal activity involves estimation of signal probability, which is the probability of a signal taking a logic value of ONE. If the primary input signal probabilities and activities are known, probabilistic or simulation based techniques [7, 9, 10, 11] can be used to estimate the activities at internal nodes of a circuit.

In this paper, the exact input stimulus for the circuit under the test mode is known. Hence, a simulation based technique can be used to obtain the actual activity at the internal nodes of a circuit. It should be observed that if accurate delay models for the logic gates and interconnects are available, then this technique will be able to accurately determine the spurious transitions at the internal nodes of the circuits. The spurious transitions causes switching of internal nodes and hence, dissipates power. The details of the power estimation technique during input reordering is given in Section 4.

The peak power dissipation is important when the circuit has to be stressed to its limit to determine its reliability. Such a situation can be considered during monitored burn-in. From Equation 1 we can observe that for a given set of stuck-at test vectors V , the condition for peak power dissipation is given by

$$\text{maximize } \left(\sum_{i=1}^n T_i^{Q_k, Q_l} C_i \right); \quad Q_k, Q_l \in V$$

where $T_i^{Q_k, Q_l}$ is given by

$$T_i^{Q_k, Q_l} = \begin{cases} 0 & \text{if } Q_k \text{ followed by } Q_l \text{ produces no transition at node } i \\ 1 & \text{if } Q_k \text{ followed by } Q_l \text{ produces a transition at node } i \end{cases}$$

The term $T_i^{Q_k, Q_l}$ is similar to activity defined in Equation 1. If one considers the nodes of a circuit to have finite delays, then glitches can occur. For such transitions, the range of values of $T_i^{Q_k, Q_l}$ can be larger than 1 based on whether no transition or a functional or hazardous (static or dynamic) transition is present at node i . Figure 2 shows an example of a hazardous transition at the output of an AND gate with unit delay model for the gates. For such input

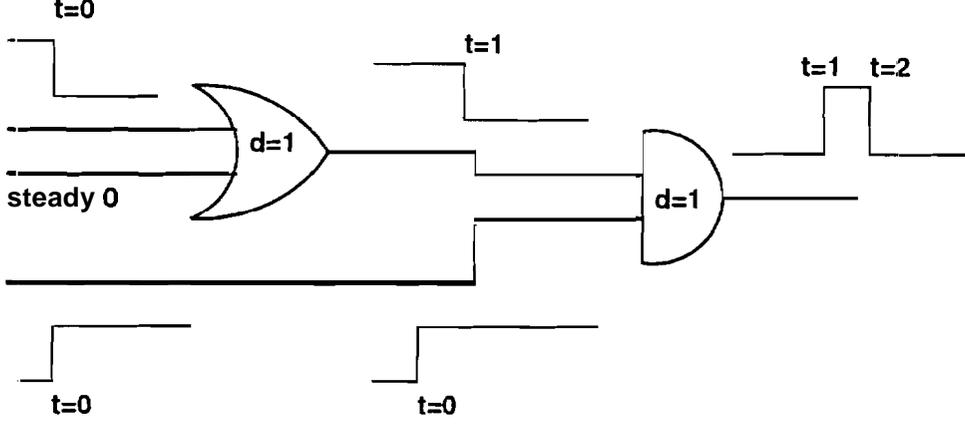


Figure 2: Spurious transition at a node

vectors, T for the output node is equal to 2. With zero delay model, T for the output will be 0.

3 Graph Based Formulation of the Problem

The problem of reordering test vectors to achieve a given power dissipation is formulated as a graph traversal problem for efficient solution. A set of test vectors V of size m is considered for a circuit C under test. Let $G(V, E, W)$ represent a graph whose set of nodes are given by V . The cardinality of set V is m , which is the number of test vectors to be applied to circuit C . E is the set of edges in the graph connecting different nodes, while W represents the set of weights associated with each edge. For example, an edge connecting nodes $Q_i \in V$ and $Q_j \in V$ is given by $e_{ij} \in E$. There are $m(m-1)/2$ edges in such a graph. Each edge is also associated with a weight $w_{ij} \in W$. Let us also assume that the graph is fully connected, i.e. there exists an edge between every pair of vertices Q_i and Q_j , ($i \neq j$). Such graphs are referred to as cliques. The weight w_{ij} associated with each edge e_{ij} is determined as follows:

$$w_{ij} = \sum_{k=1}^n T_k^{Q_i, Q_j} C_k \quad (2)$$

where C_k is the capacitive loading associated with node k of circuit C having n number of nodes. Hence, w_{ij} is proportional to the power dissipation associated with application of input vectors Q_i and Q_j , respectively.

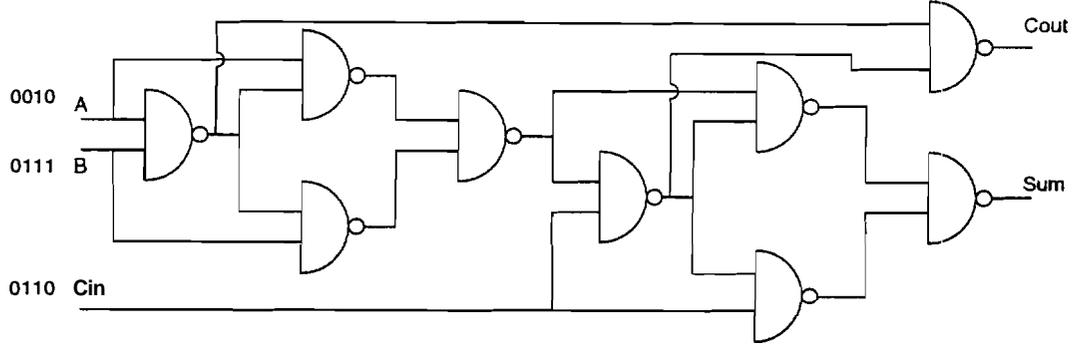


Figure 3: Full adder

For graph $G(V, E, W)$, each node k represents a test vector Q_k for a CMOS circuit C , and each edge e_{k_j} represents the current flow or power dissipation when vectors Q_j is applied after the application of vector Q_k . Hence, a Hamiltonian path of the graph G starting from node Q_1 and ending in Q_m ($Q_1, Q_2, Q_3, \dots, Q_m$) represents an ordering of test vectors for which the average power dissipation is proportional to

$$P = \frac{1}{m} \sum_{i=1}^{m-1} w_{i,i+1}$$

This suggests that the maximum or minimum length Hamiltonian path will produce an ordering of test vectors which will dissipate maximum or minimum average power. The average power dissipation is given by the summation of the edge weights of the path divided by the number of vectors. Let us consider an example. Figure 4 shows the graph G for the four vectors applied to a 1-bit full adder of Figure 3. The weight associated with each edge represents the actual simulation result of applying two test vectors to the adder circuit. For example w_{12} represents the switching activity when test vector 2 is applied after application of vector 1. The capacitance associated with each node of the full adder is approximated by the number of fanout at that node. Glitches were neglected. From Figure 4 it is clear that the maximum Hamiltonian path is Q_1, Q_2, Q_3 , and Q_4 . Hence, such a sequence of test vectors will produce the maximum average power dissipation. While the minimum Hamiltonian path, represented by Q_2, Q_4, Q_1 , and Q_3 produces minimum average power dissipation.

The problem of determining the maximum or a minimum Hamiltonian path or a path of length close to a specified value, is similar to the traveling salesman problem for which there

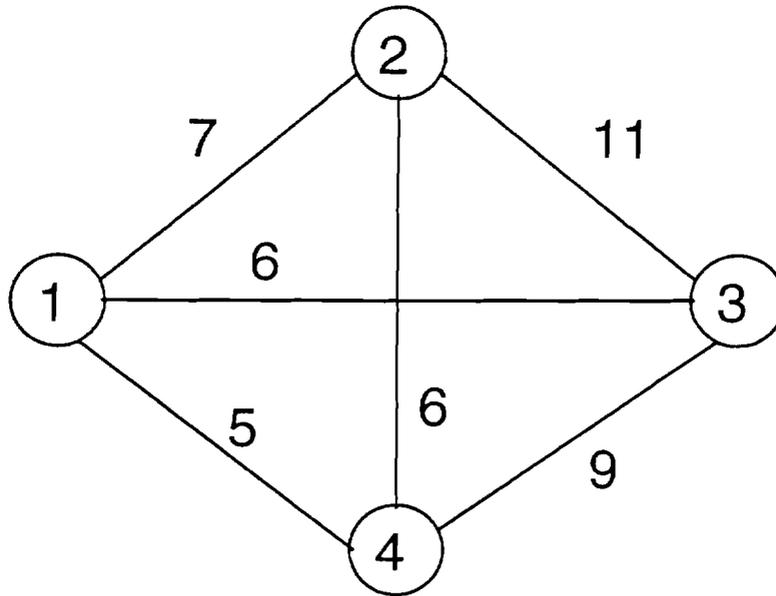


Figure 4: Graph G for the 4 test vectors of the circuit in Figure 3

exists no known polynomial time algorithm [12]. Hence, we resort to approximate methods of solution.,

3.1 Reordering for Maximal or Minimal Activity

The graph $G(V,E,W)$ can be represented by an $m \times m$ matrix M such that the (i,j) th entry M_{ij} is equal to e_{ij} . M_{ii} is undefined. We use a greedy algorithm which is able to come up with a close to optimal solution. The

sketch of the greedy procedure to determine maximum average power dissipation is given below. A very similar procedure can be used for minimization of average power dissipation.

Algorithm 1:

```

From StartNode = 1 to m {
  Generate matrix  $M$ ;
  Power = 0;
  MaxPower = 0;
  i = StartNode;
  Repeat m times: {
    Select largest  $M_{ik}$  from row  $i$ ;

```

```

    Power = Power +  $M_{ij}$ ;
    Delete row  $i$  and column  $j$  from  $M$ ;
     $i = k$ ;
}
MaxPower = Power, if Power > MaxPower;
}

```

Starting from a row r (test vector or node) of matrix \mathbf{M} the greedy algorithm selects the largest (smallest) entry in that row. That takes time $O(m)$. The row (r) and column (c) corresponding to the maximum (minimum) entry is removed from matrix M . Note that c represents the next node visited from node r . Row c is selected next for the above operation. The process is repeated m times to determine a Hamiltonian path starting at node *StartNode*. The entire process is again repeated with a different starting node for the Hamiltonian path. In the worst case, the algorithm takes $O(m^3)$ time.

3.2 Reordering for Desired Activity

In order to achieve a desired circuit activity by test reordering, a procedure similar to the one described above can be used. If the desired activity is D , then we define the *average* activity per node, $D_a = D/m$, where m is the number of nodes of graph $G(V, E)$. Instead of selecting the largest M_{ik} , as shown in Algorithm 1, the value of M_{ik} , which gives the closest running average to D_a is chosen. This approach leads to reordering such that activity is very close to the desired value. In case the desired value of activity is higher or lower than the maximal or minimal value as obtained Section 3.1, the algorithm will return the maximal or the minimal value, respectively. A sketch of the algorithm is given below.

Algorithm 2:

```

From StartNode = 1 to  $m$  {
    Generate matrix  $M$ ;
    Power = 0;

```

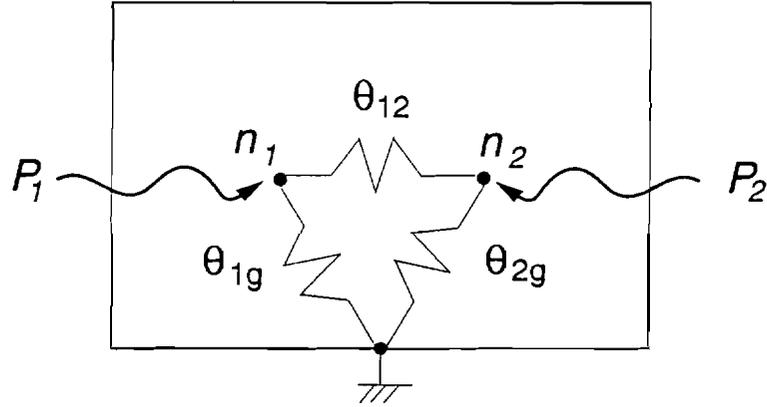


Figure 5: Model for analysis of thermal stress in a chip

```
DesiredPower = LargeNegative;
```

```
i = StartNode;
```

```
Repeat j = 1 to m times: {
```

```
  Select  $M_{ik}$  from row  $i$  such that:
```

```
   $\frac{(Power+M_{ik})}{3} - D_a$  is the smallest;
```

```
  Power = Power +  $M_{ij}$ ;
```

```
  Delete row  $i$  and column  $j$  from  $M$ ;
```

```
  i = k;
```

```
}
```

```
DesiredPower = Power, if  $\text{abs}(\text{DesiredPower} - D_a) < \text{abs}(\text{Power} - D_a)$ ;
```

```
}
```

3.3 Reordering for Power or Temperature Gradient

Hnatek in [3] observed that defects such as epitaxial and crystal imperfections, metallization, oxide, or junctional anomalies may require localized thermal stresses for the activation of the failure mechanism. In this section we will consider reordering of stuck-at test vectors to achieve such a stress or temperature gradient.

The thermal stress within a chip can be easily understood by looking at the model shown

in Figure 5. The following notations are used.

- n_1, n_2 nodes at which thermal differential is desired
- $\theta_{12} \equiv$ thermal impedance between n_1 and n_2
- $\theta_{1g}, \theta_{2g} \equiv$ thermal impedance between n_1, n_2 and chip package, respectively
- P_1, P_2 Power dissipated at n_1 and n_2 , respectively
- $t_1, t_2 \equiv$ temperature at n_1 and n_2 , respectively
- $t_a \equiv$ ambient temperature

Thermal impedance θ is defined as the temperature rise in degrees divided by the power dissipated [13]. The box shown in Figure 5 depicts the substrate of the chip. The two nodes n_1 and n_2 are two representative points on the chip, between which a thermal stress is desired. The thermal impedances shown in the figure can be calculated experimentally from sample wafers. Based on the thermal impedances, the temperatures at nodes n_1 and n_2 are calculated as follows:

$$t_1 = t_a + P_1 * \theta_{1g} + P_2 * \theta_{12} \qquad t_2 = t_a + P_2 * \theta_{2g} + P_1 * \theta_{12} \qquad (3)$$

Therefore, if a certain temperature difference between two nodes is desired, it can be expressed in terms of P_1 and P_2 . For simplicity, if we assume $\theta_{1g} = \theta_{2g}$, then for a given temperature difference $\Delta t = t_1 - t_2$, the desired power difference is

$$\Delta P = P_1 - P_2 = \frac{\Delta t}{\theta_{1g} - \theta_{12}} \qquad (4)$$

The model described above is a first-order approximation. The thermal dissipation in a chip does not take place at a particular point, rather it takes place over a distributed area. Also, to create thermal stress, the absolute values of temperatures is not as critical as the difference between them. As a result, absolute value of power is not very critical as far as the gradient is concerned. Moreover, rather than considering difference in temperature (and power), a ratio of power at two different zones can be considered, which is easy from an optimization point of view. Keeping this in mind, we can generalize the thermal stress by

considering zones or parts in a circuit instead of nodes. This generalization does not affect the basic goal of creating a thermal stress in a circuit.

Let us first consider the case when a chip is partitioned into two sections or parts, $Part_1$ and $Part_2$. In order to have a very sharp temperature or current gradient between $Part_1$ and $Part_2$, the activity in one part should be maximized while the activity in the other part can be minimized. The activities weighted by the corresponding capacitances in the two parts can be easily determined by applying a test vector Q_i followed by vector Q_j and by noting the number of nodes undergoing transition. Hence, one can construct two completely connected graphs $G_1(V, E, W^1)$ and $G_2(V, E, W^2)$, where the respective edges e_{ij}^1 and e_{ij}^2 between nodes Q_i and Q_j ($Q_i, Q_j \in V$) are weighted by the activity from the corresponding part. It should be noted that the number of vertices in each graph G_1 and G_2 is equal to the number of test vectors. The difference between the weights associated with edges e_{ij}^1 and e_{ij}^2 is equal to

$$\Delta_{ij} = |w_{ij}^1 - w_{ij}^2|$$

which signifies the difference in activity between the two parts, when two test vectors Q_i and Q_j are applied in sequence. Now, it is possible to construct a completely connected graph $G_\delta(V, E, W^\delta)$ such that each edge weight $w_{ij}^\delta = \Delta_{ij}$. Using the algorithms described above, it is possible to find a Hamiltonian path which is maximal, minimal, or of a specified length. The maximal Hamiltonian path corresponds to the ordering of test vectors which produces the maximal temperature gradient between $Part_1$ and $Part_2$, while the specified weight produces the specified temperature gradient between the parts.

Let us consider the more general case when a circuit is partitioned into p partitions. It is possible to generate graphs $G_1(V, E, W^1), G_2(V, E, W^2), \dots, G_p(V, E, W^p)$ for each of the parts such that the weight w_{ij}^k associated with edge e_{ij}^k for part $Part_k$ is equal to the activity in the part due to the application of test vectors Q_i followed by Q_j . Each part k can be associated with a stress weight s_k which specifies the relative stress that each of the parts are required to experience. For example, in case of the two-way partitioning described above,

$s_1 = 1$ and $s_2 = -1$ for maximum temperature gradient from part 1 to part 2. The edge weights in each graph $G_k(V, E, W^k)$ is multiplied by s_k to include the relative stress weighting factor into account. The new edge weight nw_{ij}^k is given by

$$nw_{ij}^k = w_{ij}^k * s_k$$

A modified graph $G_\delta(V, E, W^\delta)$ is constructed such that weight of each edge is given by

$$w_{ij}^\delta = w_{ij}^1 + w_{ij}^2 + \dots + w_{ij}^p$$

The maximum Hamiltonian path in graph $G_\delta(V, E, W^\delta)$ produces the near optimum stress or temperature gradient required across the different parts.

The methodology to partition the chip to achieve proper temperature gradient during burn-in is dependent on the chip layout. From the chip layout, the reliability engineer determines the sections of the chip across which high temperature gradient is required. Based on such information, it is possible to determine the logic gates in each partition.

4 Activity simulator

From the above discussions it is clear that efficient ways to calculate circuit activity is required to determine the the temperature or current stress across different pars. We have developed an activity simulator called *actsim*. It is an event-driven logic simulator that calculates the activity or transitions at every node within a given circuit. *Instantaneous node activity* is defined to be the number of times a node makes a zero to one or one to zero transition in the course of application of a test vector pair. In zero-delay simulation, each gate is assumed to react instantaneously to a change in its input values. Hence, there can be at most one transition recorded at each node during application of a test vector pair. To record glitching activity, the simulator can also simulate finite gate delays. In such a simulation, more than one transition may be recorded at each node, the multiple transitions arising out of internal circuit glitching activity. The instantaneous node activity at each node is multiplied by the fanout degree of the node. This is because larger capacitances are

associated with these nodes and therefore these consume proportionately more power when switched. Let the test sequence consist of the vectors Q_1, Q_2, \dots, Q_m . The instantaneous power consumed at node x when the test vector pair Q_k, Q_{k+1} is applied, is approximated by $T_x^{Q_k, Q_{k+1}} * fanout_x$. In the above, $T_x^{Q_k, Q_{k+1}}$ is the number of transitions recorded at node x and $fanout_x$ is the fanout degree of node x . The total activity cost for a circuit with n nodes and a given sequence of m test vectors is computed as

$$\sum_{k=1}^{m-1} \sum_{x=1}^n T_x^{Q_k, Q_{k+1}} * fanout_x \quad (5)$$

With regard to the graph formulation of the problem discussed in Section 3, the weight w_{ij} is given simply by

$$w_{ij} := \sum_{x=1}^n T_x^{Q_i, Q_{i+1}} * fanout_x \quad (6)$$

with $Q_k = Q_i$ and $Q_{k+1} = Q_j$. In the event that internal glitches are not considered (under zero gate delay), it is easily shown that w_{ij} does not depend on whether Q_i is applied first or vice versa. Hence the graph of Section 3 can be modeled as consisting of undirected edges. This is not necessarily true when internal glitching activity is considered. In that case the graph must be modeled with directed edges. In this paper, for simplicity, we have used the undirected graph model of Section 3 to generate the best ordering of test vectors and have only used glitching activity to verify the total costs of the optimally ordered and unordered test sequences.

The simulation proceeds from a all-zero state at every node of the circuit. As the input vectors are being applied, the changes in values at various nodes are noted and counted as activity at that node. The circuit description is given as a netlist of basic logic gates, e.g. NAND, NOR, AND, OR, and INV. The simulator can handle circuits with assignable delays and puts the resultant activity at the proper time on the timing wheel. The activity calculation is as computationally expensive as logic simulation with explicit delays.

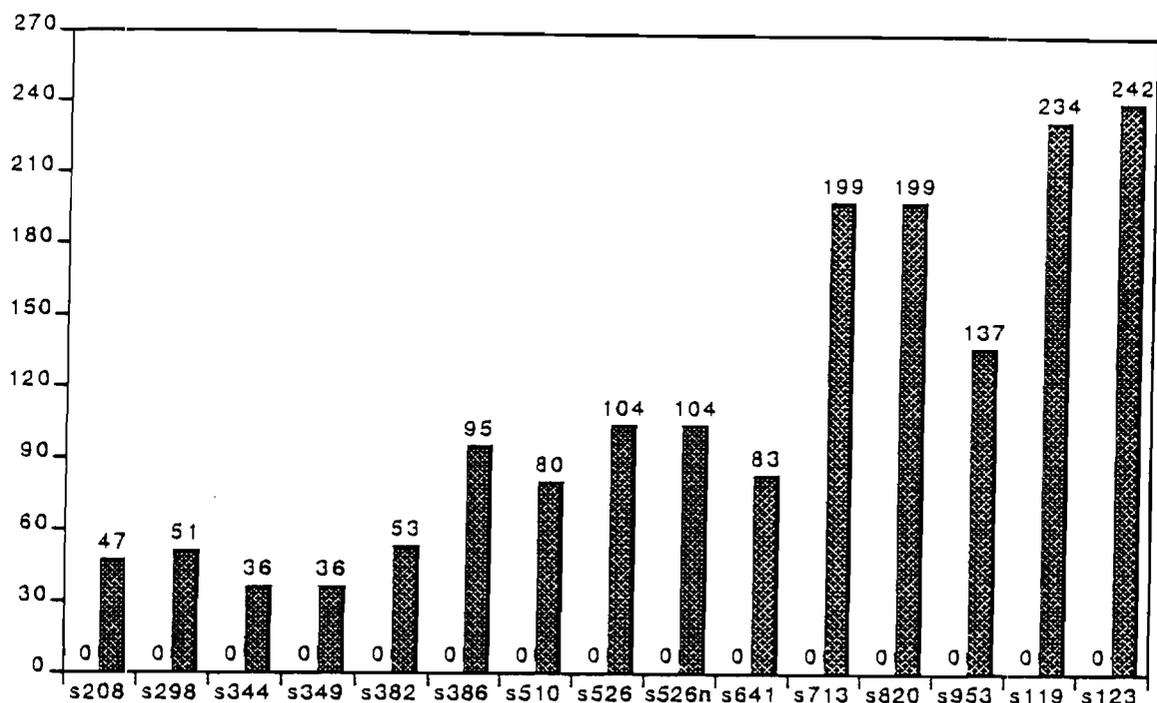


Figure 6: Test lengths for 100% test efficiency

5 Implementation and Results

The algorithms described earlier were all implemented in C on SUN workstations. The circuits used for experimentation are full-scan version of a subset of the ISCAS-89 benchmark suite. The circuits are known to be real examples from industry. The circuits were run through an automatic test generator, TRAN [14], to get test patterns with 100 percent test efficiency. These test patterns were the target for reordering such that a controlled power dissipation or the required stress or temperature gradient could be achieved. Figure 6 shows the test lengths of the circuits for 100 percent test efficiency.

Table 1 contains the minimum and maximum power drawn by the circuit for different orders of the input patterns. The column ratio shows the ratio between maximum and minimum power requirements. The ratio ranges from 2.56 to 4.31. Thus the power requirement of a circuit during application of vectors can be changed significantly by reordering of the inputs, For example, for the circuit s713, the power drawn can increase by as much as 350

Table 1: Results

Circuit	Number Vectors	Min Power	Max Power	Ratio
s27	10	23	86	3.74
s208	47	643	2736	4.26
s298	51	1877	5465	2.91
s344	36	1618	4012	2.48
s349	36	1636	4027	2.46
s382	53	2792	6858	2.46
s386	95	3304	12342	3.74
s510	80	3645	12853	3.53
s526	104	8382	22886	2.73
s526n	104	8287	22643	2.73
s641	83	5104	13042	2.56
s713	199	1355	4848	3.58
s820	199	14501	62481	4.31
s953	137	12516	33070	2.64
s1196	234	30183	86775	2.87
s1238	242	32281	100038	3.10

percent when proper reordering is achieved. During monitored burn-in, the order of test vector realizing the maximum power requirement might be the desired target.

Figure 7 graphically shows the comparison between minimum and maximum power drawn by a circuit by reordering of the input patterns. It is interesting to see that in certain situations 300% increase in power drawn can be achieved by reordering of the: input patterns.

Table 2 shows the results of specifying a desired power dissipation by reordering the input stuck-at test vectors. Such specification of power dissipation might be required in burn-ins requiring controlled power or heat generation. The results, based on Algorithm 2 shows that it is possible to control power dissipation within 5% of the desired value. However, it should be noted that if the desired power value of a circuit is outside the maximal or the minimal range of Table 1, the algorithm returns the maximal or minimal value respectively.

Figure 8 and 9 show the schematic and layout of a 4x4 multiplier. Test vectors were generated for 100% coverage of stuck-at faults. The test vectors were reordered to generate a large temperature gradient between two halves of the circuit. The layout was partitioned into 2 halves. Table 3 shows the ratio of power dissipated in one half compared to the other

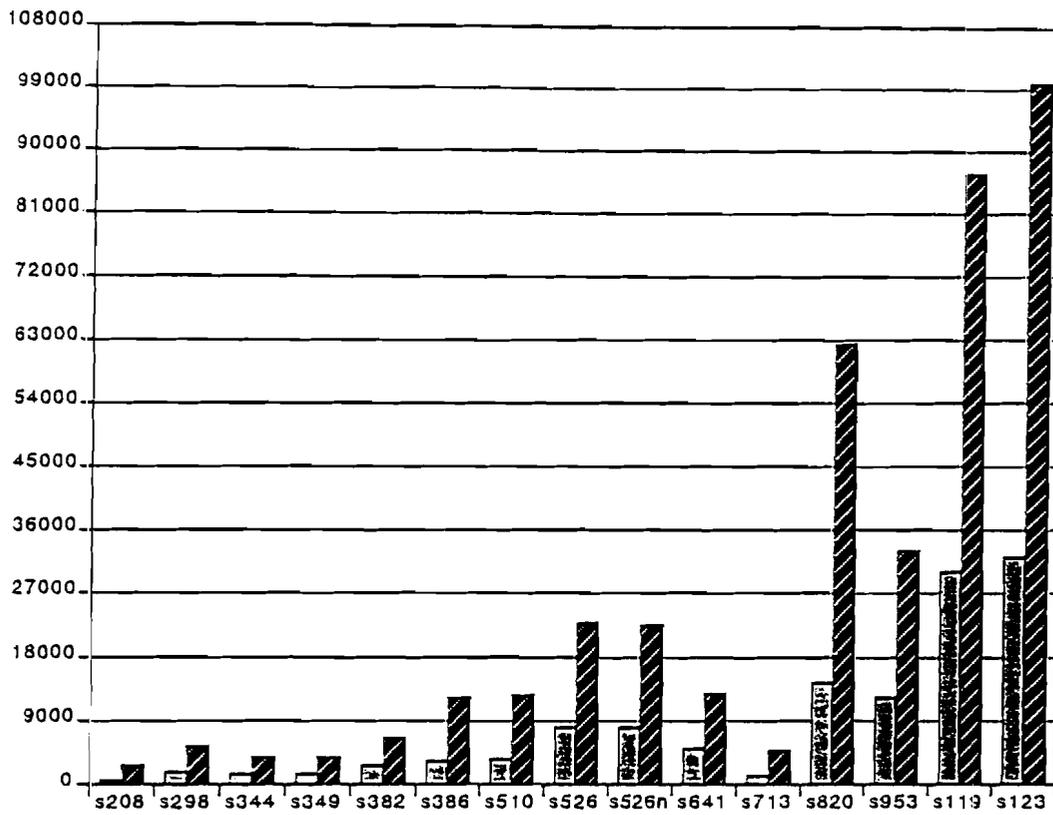


Figure 7: Minimum and maximum power drawn by circuit;

Table 2: Comparison of desired and actual power values

Design Name	Desired Power	Actual Power
s208	800	861
	1700	1699
	2100	2082
s298	3000	3001
	4000	3965
	5000	4812
s510	5000	5099
	7000	7000
	10000	9951

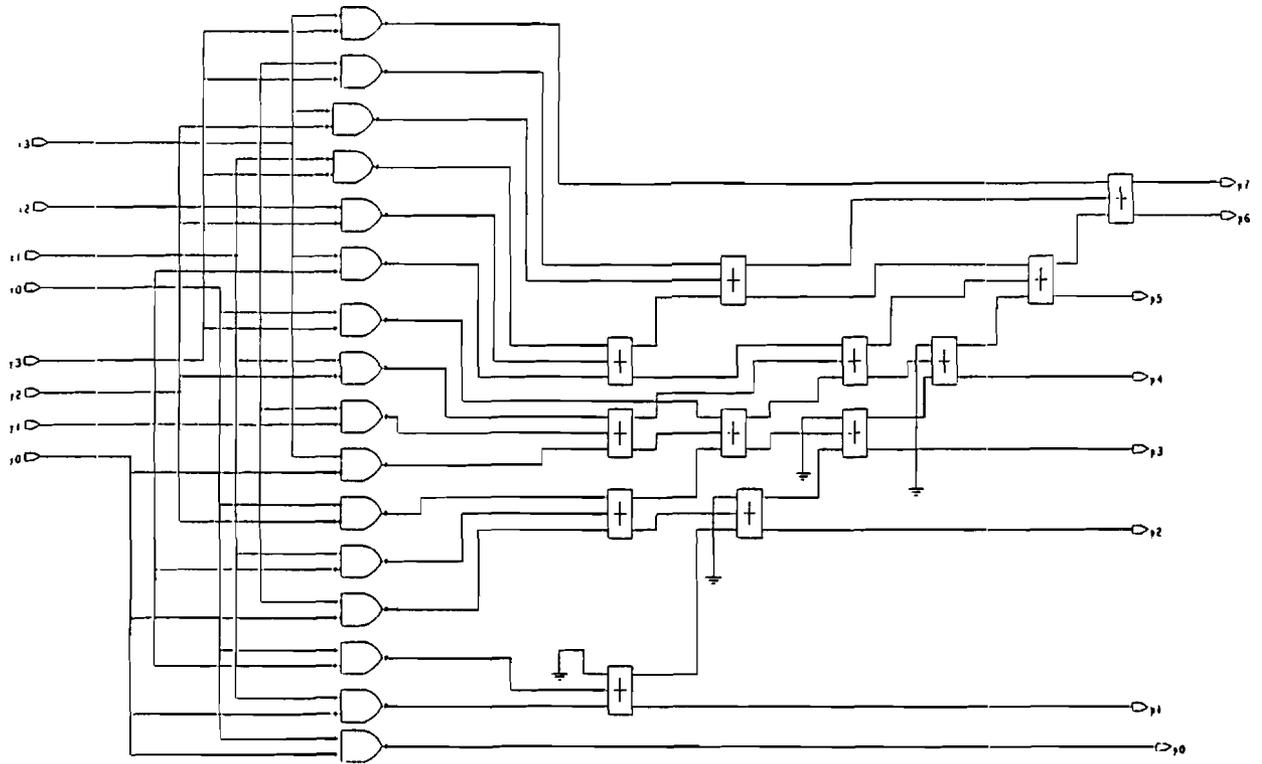


Figure 8: A 4x4 multiplier schematic

Table 3: Power or temperature gradient results for the multiplier example

Partition	Power dissipation ratio
Part1	10.69
Part2	9.10
Part3	5.12

for three different partitions of Figure 9. Results show that a large power dissipation or temperature gradient can be produced based on the stuck-at test vectors of the circuit.

6 Discussions

As semiconductor manufacturers progress towards in-situ monitored burn-in to improve quality, the dividend of this process is being reaped by increased reliability. The chips having a higher probability of infant mortality, or the so-called *borderline* devices, can be easily weeded out by proper burn-in. One issue in monitored burn-in is to be able to control the power dissipation so that the chip can be stressed as desired. While burn-in might also

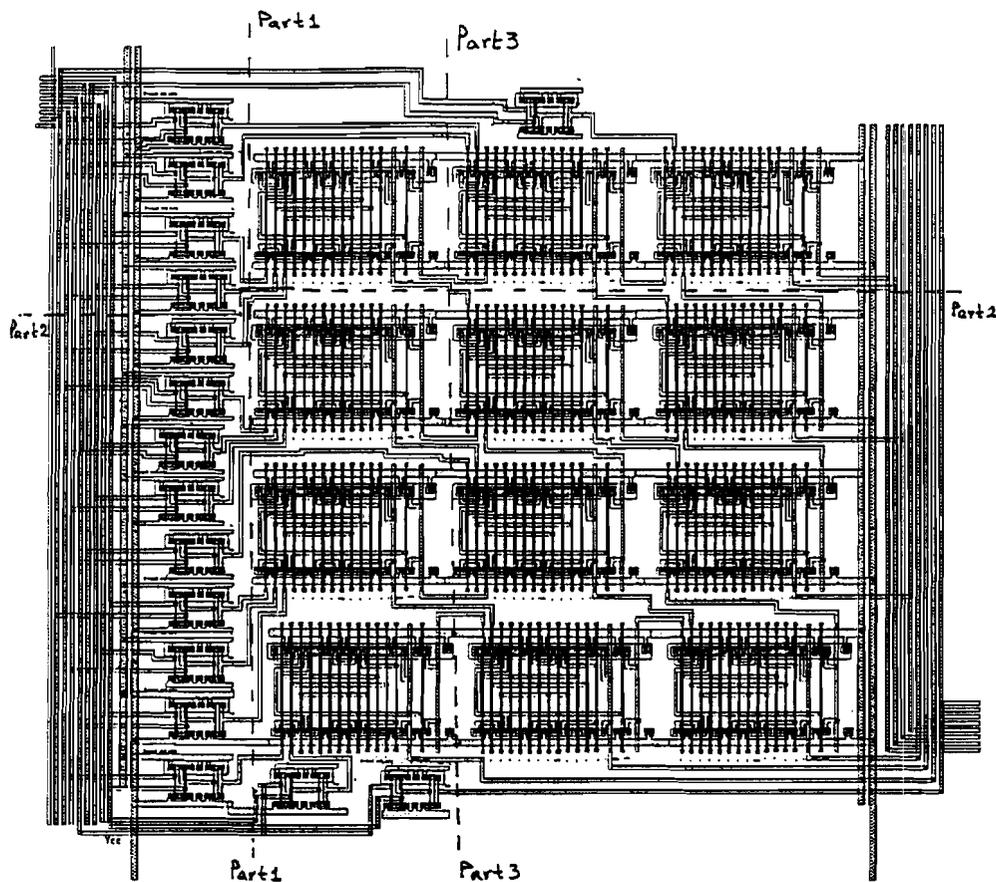


Figure 9: A 4x4 multiplier layout

require steep temperature gradient across the chip to detect certain reliability defects. Our algorithm can efficiently handle such stresses for burn-in. The methodology is very simple, and utilizes existing test patterns. By proper reordering, the number of transitions at the various inputs of gates within a circuit can be changed, thus resulting in a change in power. Our framework of power control during test application has been extended to situations where a test reordering can be found such that the power drawn by the chip will lie within specified limits. This is a very powerful methodology and we are applying it to real industrial circuits at the time of writing this paper.

References

- [1] M. Campbell, "Monitored Burn-In (A Case Study for In-Situ Testing and Reliability Studies)," *IEEE Intl. Test Conference*, 1984, pp. 518-523.
- [2] A. van den Heuvel and N. Khory, "A Rational Basis for Setting Burn-In Yield Criteria," *IEEE Intl. Test Conference*, 1984, pp. 524-530.

- [3] E. Hnatek, "Thoughts on VLSI Burn-In," *IEEE Intl. Test Conference*, 1984, pp. 531-535.
- [4] R. Iyer, D. Rossetti, and M. Hsueh, "Measurement and Modeling of Computer reliability as Affected by System Activity", *ACM Trans. on Computer Systems*, Vol. 4, No. 3, August, 1986, pp. 214-237.
- [5] R. Chou, K. Saluja, and V.D. Agrawal, "Power Constraint Scheduling of Tests," *IEEE Intl. Conf. on VLSI Design*, 1994, pp. 271-274.
- [6] A. Chandrakashan, S. Sheng, and R. Brodersen, "Low Power CMOS Digital Design," *IEEE Journal on Solid-State Circuits*, Vol. 27, No. 4, Apr. 1992, pp. 473-484.
- [7] K. Roy and S. Prasad, "Circuit Activity Based Logic Synthesis for Low Power Reliable Operations," *IEEE Trans. on VLSI Systems*, Dec. 1993, pp. 503-513.
- [8] A. Chatterjee and R. Roy, "Synthesis of Low Power Linear DSP Circuits Using Activity Metrics," *IEEE Intl. Conf. on VLSI Design*, 1994, pp. 265-270.
- [9] F.N. Najm, "Transition Density, A Stochastic Measure of Activity in Digital Circuits," *ACM/IEEE Design Automation Conf.*, 1991, pp. 644-649.
- [10] R. Burch, F. Najm, P. Yang, and T. Trick, "A Monte Carlo Approach for Power Estimation," *IEEE Trans. on VLSI Systems*, Mar. 1993, pp. 63-71.
- [11] A. Ghosh, S. Devadas, K. Keutzer, and J. White, "Estimation of Average Switching Activity in Combinational and Sequential Circuits," *ACM/IEEE Design Automation Conf.*, 1992, pp. 253-259.
- [12] M. Garey and D. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W.H. Freeman and Company, New York, NY, 1979.
- [13] P. Horowitz and W. Hill, *The Art of Electronics*, Cambridge University Press, New York, NY, 1989.
- [14] S. T. Chakradhar, V. D. Agrawal, and S. G. Rothweiler, "A Transitive Closure Algorithm for Test Generation," *IEEE Trans. CAD*, July 1993, pp. 1015-1028.