Test Power Reduction by Simultaneous Do not Care Filling and Ordering of Test Patterns Considering Pattern Dependency

T. Sarkar*, S. Nath Pradhan

Department of ECE, National Institute of Technology, Agartala. Agartala, India

1. INTRODUCTION

Reduction of power consumption at the time of circuit testing becomes a challenging issue as the design consumes more power in test mode compared to functional mode of operation [1]. At present, testing is one of the most vital issues in the development process of an integrated circuit. The issues that depend on testing are manufacturing yield, test cost and product quality [2-4]. It was shown that test power is much higher than the power consumption in normal functional mode due to several reasons such as (i) Automatic test pattern generation (ATPG) tools generate test patterns that have high toggle rate in order to reduce pattern count and test application time. Therefore the switching activity of the circuit in test mode is often several times compared to normal mode of operation. (ii) To reduce test application time parallel testing is also used, mainly for System-on-Chip (SOC) devices. This parallelism increases power dissipation during test. (iii) The Design-for-Testability (DFT) circuitry is inserted in the circuit under test to improve testing issues. DFT circuit remains idle at normal operation but is used in test mode. These additional active elements further increase power dissipation. In this work single stuck-at fault model is considered. In this fault model the value on the faulty signal line appears to be stuck either at logic '0' or logic ‘1’, referred to as stuck-at-0 or stuck-at-1 respectively. Test patterns are generated using ATALANTA tool. CAD tool- ATALANTA generates test patterns targeting stuck-at fault.

Most of the works related to power reduction in the literature describes about dynamic power minimization. But as the technology shrinks down below 65nm leakage power dominates over dynamic power.

*Corresponding Author Email: trupa.sarkar@gmail.com (T. Sarkar)
leakage power has become an important task in overall power minimization of the circuit. It is considered that leakage power depends only on the current input pattern applied to the circuit. But runtime leakage power [5] depends on both the previous and present input pattern applied to circuit [6] and this runtime leakage power also changes with the change of test application time. So, leakage power depends not only on the order of test patterns applied to the circuit but also on the applied time period. Also dynamic power depends on the switching activity between test patterns fed to the test circuit. So, dynamic power depends on the ordering of test patterns applied to the circuit. In this paper a method has been proposed based on Genetic Algorithm (GA) to solve the integrated problem for don’t care filling of test patterns and to reorder the applied test vectors so that the transition density between the consecutive test vectors are minimum which further decreases the leakage power consumption without compromising the fault coverage.

This paper is structured as follows. Section 2 deals with the related work on power minimization and gives an insight to the calculation of runtime leakage power. Section 3 describes proposed algorithm for don’t care filling. Section 4 enumerates the experimental results. Conclusion is given in section 5.

2. RELATED WORK

Badereddine et al. [7] proposed a method in which the X-bits of test patterns are assigned such that it reduces the number of bit transitions in consecutive test patterns which lowers scan cell transitions. The X bits of the pattern assigned with adjacent 0 or 1 heuristics so that the peak power is reduced. Here only the peak power is taken into consideration. But as the technology reduces it becomes necessary to reduce the leakage power consumption.

Kumar et al. [8] studied that if the specific bits generated by ATPG tool and which are a part of the deterministic set of test vectors are identified first, they will detect the targeted fault models. The remaining bits are used as don’t care (X) bits to transform the original vector to a power aware vector. The knowledge of fault propagation path and fault activation path are also utilized here. PSO (Particle Swarm Optimization) based approach is used for vector reordering which uses Travelling Salesman Problem (TSP) concept. This proposed technique reduces both runtime leakage power and dynamic power without any change of fault coverage. The drawbacks in literature [8] have been addressed in this paper. Here PSO based technique is used to reorder the power aware vector. But runtime leakage power depends on the time period applied to the circuit which has not been considered here.

Chattopadhyay and Choudhary [9] proposed a technique based on genetic algorithm to generate a set of test patterns that minimize power dissipation at the time of testing. It also optimizes the order of selected patterns applied to CUT to reduce the switching activity of individual circuit gates under a zero gate delay model.

In the literature the authors suggested several methods for power aware X-filling of test patterns and reordering the patterns to minimize the total power consumption. But reordering the filled up pattern will disturb the optimum results obtained by don’t care filling of test vectors. Also none of the above techniques consider the runtime leakage power which depends on previous patterns and time period applied to the circuit under test for ordering of test patterns. In this work an approach based on GA is used to fill up the don’t care present in the set of test patterns and to reorder the patterns simultaneously considering leakage power dependency on previous patterns and time period such that when these patterns are applied to CUT the total power consumption is low.

2.1. Calculation of Runtime Leakage Power

A technique has been suggested in literature [10] to calculate runtime leakage power. The leakage of logic gate depends on previous pattern and as well as on the time period. For example, the effect of leakage power on previous pattern at different time period for 2-input NAND gate (Figure 1) is shown in Table 1. When the previous pattern is 01, the internal node capacitor of capacitance Cm is totally discharged. After 10ns the when the pattern is changed to 00 from 01, a small leakage current begins to charge node capacitor. Due to staking effect the leakage current through M1 drops as Vm rises. The large turn on current through M1 starts charging Cm and leakage current continues to charge Cm even when M1 turns off. If M1 turns on by the change of input from 00 to 01 Cm discharge quickly and leakage transition is spontaneous.

Tables 1, 2 and 3 represents the runtime leakage power for 2 input NAND gate, 2 input NOR gate and NOT gate for all the possible combinations of input patterns.
A new chromosome (offspring) is created by combining two chromosomes to produce a new chromosome (offspring). The population size is kept uniform throughout the generations.

**Chromosome structure** - Let the test patterns generated by the ATALANTA for C17.bench circuit are ‘100xx11’ (1st pattern), ‘00xxx1x’ (2nd pattern), ‘xx101xx’ (3rd pattern), ‘0xx1x11’ (4th pattern), ‘1xx10111’ (5th pattern), ‘00x1x10’ (6th pattern) and ‘001xx01’ (7th pattern). Here, ‘x’ represents don’t care. The number of don’t cares present in the set of test pattern is 19 and total number of test patterns generated is 7. The chromosome consists of two parts. One is ‘don’t care filling’ part which contains all the don’t cares that exist in the generated test patterns. The other part of chromosome is the ‘ordering of test patterns’ part which contains the number of test patterns generated by ATALANTA. Therefore the chromosome will look like:

Don’t care filling of test patterns ordering of test patterns

<table>
<thead>
<tr>
<th>Pattern</th>
<th>Don't Care Filling</th>
<th>Order of Patterns</th>
</tr>
</thead>
<tbody>
<tr>
<td>1001101011010001011</td>
<td>5427163</td>
<td></td>
</tr>
</tbody>
</table>

Here the 5th pattern is applied first followed by 4th pattern, 2nd pattern and so on as seen in the chromosome structure.

**Step 1: Generation of initial population** - The first don’t care of the 5th pattern is substituted by the first bit of chromosome. Correspondingly the next bit of the chromosome will replace the second X-bit of the next pattern and this continues until it reaches the last bit of chromosome.

**Step 2: Calculation of fitness** - Each chromosome represents one set of test pattern. As, we are targeting to minimize power without affecting fault coverage the cost function or fitness will consist of leakage and dynamic power.

\[ \text{Cost} = P_{\text{leakage}} + P_{\text{dynamic}} = \alpha C_L V_{dd}^2 f + I_{\text{leak}} V_{dd} \]

where, supply voltage is \( V_{dd} \), \( C_L \) represents load capacitance, \( \alpha \) is the switching activity, \( f \) is the frequency of operation which is the inverse of critical delay of the circuit.

Value of \( C_L \) and \( I_{\text{leak}} \) for all the gates are stored in Look-up-Table (LUT) after simulation in CADENCE tool at 45nm technology and using supply voltage of 1V.

**Step 3: Genetic operator** - Here genetic operators are used to generate populations for next generation. The three operators of GA are selection, crossover and mutation.

**A. Chromosome Selection** - The chromosomes are selected from the population based on fitness value for crossover, mutation and to produce offspring.

**B. Crossover** - It combines two chromosomes to produce a new chromosome (offspring). The main idea of
crossover is to get a new chromosome which may be better than both of the parents if it takes the best characteristics from each of the parents. Here we are using two point crossover. Consider the following crossover operation.

Parent 1:

\[
\begin{array}{ccccccc}
A1 & B1 & C1 & D1 & E1 & F1 \\
1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 6 & 3 \\
\end{array}
\]

Parent 2:

\[
\begin{array}{ccccccc}
A2 & B2 & C2 & D2 & E2 & F2 \\
0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 3 & 5 & 7 & 2 & 1 & 4 & 6 \\
\end{array}
\]

After crossover the offspring produced will be

Offspring 1:

\[
\begin{array}{ccccccc}
A1 & B2 & C1 & D1 & E2 & F1 \\
1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 5 & 4 & 7 & 2 & 1 & 6 & 3 \\
\end{array}
\]

Offspring 2:

\[
\begin{array}{ccccccc}
A2 & B1 & C2 & D2 & E1 & F2 \\
0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 3 & 5 & 2 & 7 & 1 & 4 & 6 \\
\end{array}
\]

C. Mutation- 10% of the chromosomes get mutated randomly and populate in the next generation. For mutation two sets of numbers s1, s2 (for ‘don’t care filling of test patterns’ part) and r1, r2 (for ‘ordering of test patterns’ part) are randomly generated. For ‘don’t care filling of test patterns’ part all the bits between s1 and s2 and changed 1 to 0 and from 0 to 1. For ‘ordering of test patterns’ part all the individual operand (oi) within r1 and r2 get changed with the (N-oi+1), where N is the number of patterns generated.

\[
\begin{array}{cccc}
s1 & s2 & r1 & r2 \\
1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 5 & 4 & 1 & 6 & 7 & 2 & 3 \\
\end{array}
\]

Here N is equal to 7. Therefore the operand 4 will be (7-4+1=4). So the operand 4 remains same. The next operand 7 will be (7+7=14). In this case 7 is replaced by 1. After mutation the chromosome becomes

\[
\begin{array}{ccccccc}
1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 5 & 4 & 7 & 2 & 1 & 6 & 3 \\
\end{array}
\]

Step 4: Termination- The steps for GA are repeated until there is no improvement in fitness function after 10 iterations

The steps for finding low power dissipation pattern considering integrated approach for don’t care filling and ordering taking into account the previous pattern dependency using GA is summarized below:

a) An initial population is generated whose size is equal to the number of don’t care present in the test patterns and the number of test patterns generated.
b) The fitness function, of initial population is calculated according to leakage value in the LUT in Tables 1, 2 and 3.
c) For crossover a pair of parent chromosome is taken from current population based on the method described in literature [11].
d) Crossover is done by randomly chosen point to form two unique don’t care bits and test pattern ordering.
e) Mutation is done for each bit of the chromosome and a unique filled don’t care bits and test pattern ordering is generated.
f) 80% of the newly generated test patterns and 20% of the existing test pattern are put in a new population. Now fitness function for new population is calculated.
g) Repeat steps (c) to (f) for a number of generations till there is no improvement in values over last 40 generations.
h) Terminate the process and return the optimal solution.

The flowchart of proposed algorithm is given in Figure 2.

4. EXPERIMENTAL RESULTS

Here we have considered ISCAS’85 benchmark circuits for our experiments. Test patterns with maximum fault coverage are generated with –D 1 option of ATALANTA. The existing don’t cares in the test patterns are filled and reordered simultaneously using GA considering runtime leakage power dependency on previous patterns and time period without effecting fault coverage. Runtime leakage power for 2 input NAND gate, 2 input NOR gate and NOT gate is calculated considering pattern dependency in ‘Cadence Virtuoso Analog Design Environment’ at 45 nm technology using supply voltage (Vdd) of 1 volt. The proposed method is coded in C language. To reduce the size and complexity of the Look up Table (LUT) all the circuits are mapped into 2-input NAND gate, 2 input NOR gate and NOT gate is calculated considering pattern dependency in ‘Cadence Virtuoso Analog Design Environment’ at 45 nm technology using supply voltage (Vdd) of 1 volt. The proposed method is coded in C language. To reduce the size and complexity of the Look up Table (LUT) all the circuits are mapped into 2-input NAND gate, 2-input NOR gate and NOT gate is calculated considering pattern dependency in ‘Cadence Virtuoso Analog Design Environment’ at 45 nm technology using supply voltage (Vdd) of 1 volt. The proposed method is coded in C language.
Except the first column the other column consists of three sub columns. The leakage power consumption for test patterns after don’t care filling and with default ordering is shown in the first sub column. The second sub column represents the results for test patterns after don’t care filling followed by ordering. The third sub column shows the results of proposed method for simultaneous don’t care filling followed by ordering of test patterns. The third sub column shows the % of leakage power savings after simultaneous don’t care filling and ordering of test patterns. Table 5 shows the % of leakage power savings after simultaneous don’t care filling and ordering of test patterns using GA compared to test patterns with default ordering. For C432 circuit the savings is 66.40% at 40ns and 66.36% at 100ns. Table 6 shows the percentage of leakage power savings after simultaneous don’t care filling and ordering of test patterns at different time period compared to test patterns with don’t care filling followed by ordering. The cumulative power savings is 7.32% at 40ns and 7.39% at 100ns for C432 circuit. Table 7 shows the dynamic power for simultaneous ordering and filling of test patterns at different time period. It is observed that dynamic power also decreases using the proposed method. Table 8 shows the total power (dynamic and leakage power) consumption for test patterns after don’t care filling with ordering and after application of the integrated approach for don’t care filling and ordering of test patterns at four different time period. From Table 9 it can be seen that for C432 circuit the total power (dynamic and leakage power) savings is 9.02% at 40ns and 9.18% at 100ns.

From the results the following observations can be made.

1. From Table 4 it is observed that test patterns with don’t care filling and with default ordering consumes maximum leakage power and as the time period increases this leakage power dependency on the ordering of the patterns gradually decreases. Also in general it is observed that as the time period increases the difference in the leakage power consumption compared to the previous time period decreases.

2. From Table 5 it is seen that compared to default ordering the leakage power savings is maximum for C432 circuit which is 66.40% at 40ns.

3. From Table 6 the maximum leakage power savings takes place for C1908 circuit at 100ns compared to don’t care filling followed by ordering of test patterns.

4. From Table 9 it is observed that the maximum savings in total power consumption is 14.25% for C1908 circuit at 100ns.

From the above observations it can be concluded that the integrated approach for don’t care filling and ordering of test patterns gives better results compared to don’t care filling with default ordering of test patterns and don’t care filling followed by ordering of test patterns.

---

**TABLE 4.** Cumulative leakage power for simultaneous ordering and filling of test patterns at different time period

<table>
<thead>
<tr>
<th>Circuit</th>
<th>10ns</th>
<th>20ns</th>
<th>40ns</th>
<th>100ns</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>before ordering of test patterns (nW)</td>
<td>with ordering of test patterns (nW)</td>
<td>Proposed simultaneous ordering and filling of test patterns (nW)</td>
<td>before ordering of test patterns (nW)</td>
</tr>
<tr>
<td>c432</td>
<td>7735.7</td>
<td>2770.8</td>
<td>2610.8</td>
<td>7353.7</td>
</tr>
<tr>
<td>c880</td>
<td>193749.9</td>
<td>105312.1</td>
<td>103171.0</td>
<td>185979.4</td>
</tr>
<tr>
<td>c1908</td>
<td>44229.3</td>
<td>34447.1</td>
<td>30176.2</td>
<td>43148.4</td>
</tr>
<tr>
<td>c1355</td>
<td>276229.6</td>
<td>198527.3</td>
<td>175652.4</td>
<td>268712.6</td>
</tr>
</tbody>
</table>
**TABLE 5.** Percentage of leakage saving after simultaneous don’t care filling and ordering of test patterns at different time period compared to default ordering of test patterns

<table>
<thead>
<tr>
<th>Circuit</th>
<th>10ns</th>
<th>20ns</th>
<th>40ns</th>
<th>100ns</th>
</tr>
</thead>
<tbody>
<tr>
<td>c432</td>
<td>66.25</td>
<td>64.78</td>
<td>66.40</td>
<td>66.36</td>
</tr>
<tr>
<td>c880</td>
<td>46.75</td>
<td>46.17</td>
<td>46.76</td>
<td>47.09</td>
</tr>
<tr>
<td>c1908</td>
<td>31.77</td>
<td>30.98</td>
<td>32.03</td>
<td>32.86</td>
</tr>
<tr>
<td>c1355</td>
<td>36.41</td>
<td>48.90</td>
<td>35.47</td>
<td>36.42</td>
</tr>
</tbody>
</table>

**TABLE 6.** Percentage of leakage saving after simultaneous ordering and filling of test patterns compared to reordering of test patterns at different time period

<table>
<thead>
<tr>
<th>Circuit</th>
<th>10ns</th>
<th>20ns</th>
<th>40ns</th>
<th>100ns</th>
</tr>
</thead>
<tbody>
<tr>
<td>c432</td>
<td>5.77</td>
<td>2.46</td>
<td>7.32</td>
<td>7.39</td>
</tr>
<tr>
<td>c880</td>
<td>2.03</td>
<td>1.00</td>
<td>2.09</td>
<td>2.73</td>
</tr>
<tr>
<td>c1908</td>
<td>12.39</td>
<td>11.45</td>
<td>12.85</td>
<td>13.96</td>
</tr>
<tr>
<td>c1355</td>
<td>11.52</td>
<td>9.03</td>
<td>9.74</td>
<td>11.08</td>
</tr>
</tbody>
</table>

**TABLE 7.** Dynamic power for simultaneous ordering and filling of test patterns at different time period

<table>
<thead>
<tr>
<th>Circuit</th>
<th>10ns</th>
<th>20ns</th>
<th>40ns</th>
<th>100ns</th>
</tr>
</thead>
<tbody>
<tr>
<td>c432</td>
<td>5512.68</td>
<td>5216.66</td>
<td>5878.27</td>
<td>5363.87</td>
</tr>
<tr>
<td>c880</td>
<td>211604.04</td>
<td>203720.08</td>
<td>199508.54</td>
<td>195398.67</td>
</tr>
<tr>
<td>c1908</td>
<td>69142.04</td>
<td>60286.03</td>
<td>59507.47</td>
<td>52152.34</td>
</tr>
<tr>
<td>c1355</td>
<td>406959.94</td>
<td>354502.80</td>
<td>345482.59</td>
<td>307378.86</td>
</tr>
</tbody>
</table>

**TABLE 8.** Total power consumption for proposed method

<table>
<thead>
<tr>
<th>Circuit</th>
<th>10ns</th>
<th>20ns</th>
<th>40ns</th>
<th>100ns</th>
</tr>
</thead>
<tbody>
<tr>
<td>c432</td>
<td>5512.68</td>
<td>5216.66</td>
<td>5878.27</td>
<td>5363.87</td>
</tr>
<tr>
<td>c880</td>
<td>211604.04</td>
<td>203720.08</td>
<td>199508.54</td>
<td>195398.67</td>
</tr>
<tr>
<td>c1908</td>
<td>69142.04</td>
<td>60286.03</td>
<td>59507.47</td>
<td>52152.34</td>
</tr>
<tr>
<td>c1355</td>
<td>406959.94</td>
<td>354502.80</td>
<td>345482.59</td>
<td>307378.86</td>
</tr>
</tbody>
</table>

**TABLE 9.** Percentage of total power savings after simultaneous ordering and filling of test patterns compared to don’t care filling and ordering of test patterns at different time period

<table>
<thead>
<tr>
<th>Circuit</th>
<th>10ns</th>
<th>20ns</th>
<th>40ns</th>
<th>100ns</th>
</tr>
</thead>
<tbody>
<tr>
<td>c432</td>
<td>5.36</td>
<td>8.75</td>
<td>9.02</td>
<td>9.18</td>
</tr>
<tr>
<td>c880</td>
<td>3.72</td>
<td>2.05</td>
<td>4.75</td>
<td>5.01</td>
</tr>
<tr>
<td>c1908</td>
<td>12.80</td>
<td>12.36</td>
<td>13.95</td>
<td>14.25</td>
</tr>
<tr>
<td>c1355</td>
<td>14.22</td>
<td>11.03</td>
<td>12.09</td>
<td>11.49</td>
</tr>
</tbody>
</table>
5. CONCLUSION

In this paper an integrated approach based on genetic algorithm is used for don’t care filling and for ordering of test patterns considering pattern dependency. For our experiments we have considered ISCAS 85 benchmark circuits. Previous works on low power consumption concerns mainly about don’t care filling of test patterns or with don’t care filling and then ordering of test patterns. If reordering is done after don’t care filling then it will affect the optimum results obtained after don’t care filling. Considering the above issue we have proposed a method based on GA in which the chromosome consists of don’t care filling and ordering part. Considering the proposed method a maximum savings of 66.36% is obtained for C432 circuit compared to previous method.

6. ACKNOWLEDGEMENT

This work was supported by SMDP-C2SD project, sponsored by Deity, Govt. of India.

7. REFERENCES


Test Power Reduction by Simultaneous Do Not Care Filling and Ordering of Test Patterns Considering Pattern Dependency

T. Sarkar*, S. Nath Pradhan

Department of ECE, National Institute of Technology, Agartala, Agartala, India

Keywords: Don't Care, Test Pattern Ordering, Pattern Dependency

Paper Info

Paper history: Received 17 March 2017 Received in revised form 13 January 2018 Accepted 08 March 2018

Keywords: Testing, Don't Care, Run Time Leakage, Fault Coverage, Genetic Algorithm, Test Pattern Ordering, Pattern Dependency