Defence Science Journal, Vol. 52, No. 4, October 2002, pp. 351-356

Generation of High Power Test Vector Set for Combinational VLSI Circuits

K.T. Oommen Tharakan

Vikram Sarabhai Space Centre, Thiruvananthapuram – 695 022

and

S.S.S.P. Rao

Indian Institute of Technology Bombay, Mumbai – 400 076

ABSTRACT

Burn-in is a well-known technique that helps to accelerate failure mechanisms to surface out latent defects, which are not activated during normal testing of the VLSI devices. The devices are kept at a specified high temperature, for a specified period, in static or dynamic conditions. Since this method is cumbersome, an alternate method based on complementary metal oxide semiconductor (CMOS) signal switching for VLSI devices is considered. The majority of power dissipation in CMOS circuitry is due to the switching current associated with charging and discharging of load capacitances. Hence, if the test vectors can be so designed that maximum activity is conjured, the stress on the device can be maximised. In this paper, a new algorithm for generation of these power vectors from a gate-level description of the circuit is presented. The method has been applied to different circuits and the results compared.

Keywords: Burn-in simulation, latent defects, failure mechanisms, burn-in test vectors, stress testing, VLSI devices, VLSI circuits, CMOS signal switching, power vectors, algorithm, CMOS gates, CMOS device, complementary metal oxide semiconductor device

1. INTRODUCTION

VLSI circuits are increasingly being used in space applications due to their technological advancements in miniaturisation, resulting in higher performance and increased complexity of the integrated circuits (ICs). But the usage of these devices in space applications calls for high reliability. The design and fabrication techniques can propagate into the chip defects. Units with infant mortality problems often have latent defects, which cannot be detected during normal testing of the VLSI circuits. Stressing the circuit with high current densities and temperature can activate such defects.

1.1 Stress Testing Methodology

Burn-in stresses the circuit for weeding out infant mortality problems. Monitored burn-in is the most effective test in detecting these defects, but it requires application of test vectors during burn-in. But since these tests are very tedious, stress testing is adopted, wherein latent defects can be

Received 02 September 2002

weeded out by higher current densities or localised current stresses while achieving high fault coverage. Here, the stress test vectors are exercised on a circuit for a desired period to separate devices with infant mortality issues.

1.2 Power Dissipation in CMOS

Power dissipation in complementary metal oxide semiconductor (CMOS) circuits is due to the three main factors–leakage current, short-circuit current, and switching current associated with charging or discharging load capacitances. The latter component contributes to the majority of power in the CMOS gates. It is to be noted that computing the dissipation of a complex gate is modified by the switching activity factor $f_{0\rightarrow 1}$. While this factor can be easily computed for an inverter, it turns out to be far more complex in the case of higher-order gates and circuits. One concern is that the switching activity of a network is a function of the nature and statistics of the input signals. If the input signals remain unchanged, no switching happens and the dynamic power dissipation is zero. On the other hand, rapidly changing signals provoke plenty of switching, and hence, dissipation. Other factors influencing the activity are the circuit style, the function to be implemented and the overall network topology.

1.3 Activity Generator

Maximising the activity of a circuit is the key in generating the test vectors for burn-in simulation. All published methods for the estimation of the signal activity involve estimation of the signal probability, which is the probability of a signal taking a logic value of 1. If the primary input signal probabilities and activities are known *a priori*, the generation of the test vector could have been simplified. As this is not the case, simulation is called for. For this, test vectors are generated, such that all the primary output and the internal nodes are toggled. The target is that a 1 to 0 and a 0 to 1 transition occurs in the circuit under consideration.

2. GRAPH-BASED ACTIVITY GENERATION

The problem of re-ordering the test vectors can be considered as a graph-traversal problem. The graph consists of vertices formed by various test vectors, edges are formed by weighed Hamming distance of the logic values of all the intermediate nodes of the circuit generated by one test vector wrt other and the weight is a decimal value associated with a node based on its criticality (if any). Usually the weight is taken as 1.

Some of the key terms used in the description of the algorithm are:

- **Level of a gate:** The level of a gate is a measure of its distance from the primary input. The primary input are level zero. The level of a gate is one higher than the highest level input to this gate.
- **Current gate:** This is defined as a gate of the current level, which is presently considered for the backward drive (consistency/justification).
- **Pivotal vector:** Pivotal vector (PV) is the first vector, based on which the next maximum activity vector is computed using the algorithm.
- **Propagating power vector set:** Propagating power vector set (PPVS) is the set of test vectors in that order which may result in maximum activity of the circuit under consideration. PPVS will have PV as the first element.
- **Power vector set:** Power vector set (PVS) is the set of test vectors in that order which results in maximum activity of the circuit under consideration. This is basically the ordered PPVS.
- **Non-repeating vector:** Non-repeating vector (NRV) is an input vector, which will occur only once in PVS.
- **Weighed Hamming distance:** The weighed Hamming distance of a vertex of a graph wrt another vertex is the Hamming distance of the internal logic states of all the nodes of the circuit created by the corresponding test vectors wrt the same created by a test vector of the adjacent vertex. In calculating the weighed Hamming distance based on the criticality of the node of a circuit, weights can be added.

2.1 Rules of Algorithm

The two rules have been formulated, which aid in the completion of the algorithm to follow. The first rule aids in test generation and the second in ordering the test vectors.

Rule 1: If the current gate output is not a primary output, find the input vectors targeting the logical state of the current gate, such that any of the primary output gets toggled. If the current gate is a primary output, backtrack to find the input vectors. The input vectors are generated using D-algorithm or PODEM.

Rule 2: Compute the PVS as the set of test vectors corresponding to the vertices of the maximum spanning tree of the graph, in that order in which the tree is traversed.

2.2 Algorithm

A high level description of the algorithm is appended below:

```
Levelise the circuit
Select a pivotal vector
Do logic simulation of the circuit
begin
  current level = maximum level
    repeat
      for all gates in the current level
      do
        complement the logical state of the
        current gate
        find the NRV with rule1.
        If  (NRV≠Ø) then
        begin
        Attach this NRV as the next element
        of PPVS
        compute the logical state of
        all nodes with this NRV
        compute the weighed Hamming
        distance of the edges formed with
        previous vertices
        else
        update current gate with the next gate
        end
      od
    decrement current level by 1
  until (current level = first level)
  end
  Obtain the PVS using rule 2
```

2.2.1 *Algorithm Illustrated*

The algorithm has been illustrated using a typical circuit (Fig. 1).

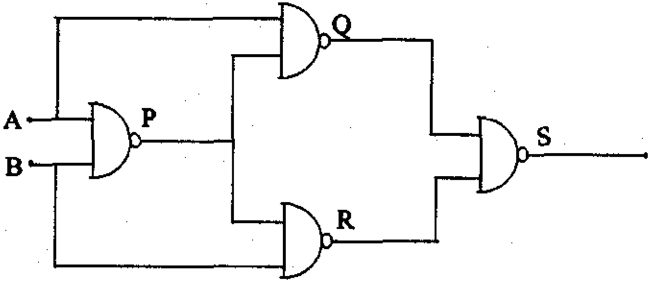


Figure 1. Circuit 1

- PV is 11.
- PPVS = {11}
- Level = 3

  NRV = {10}

  PPVS = {11,10}
- Level = 2

  NRV = {00}

  PPVS = {11,10,00}

  NRV = {01}

  PPVS = {11,10,00,01}
- Level = 1

  NRV = Ø (since the vector computed 11 is not an NRV)

  PPVS = {11, 10, 00, 01}

Next, one has to construct the maximum spanning tree of the graph with nodes A, B, C, D with logical output 011, 101, 111, and 110.

The graph is given in Fig. 2. The vertices represent the test vectors (given in the square brackets) generated using *rule* 1. The logical values of the internal nodes corresponding to each test vector are given in the curved brackets. The edges represent the weighed Hamming distance of each vertex wrt its adjacent vertex.

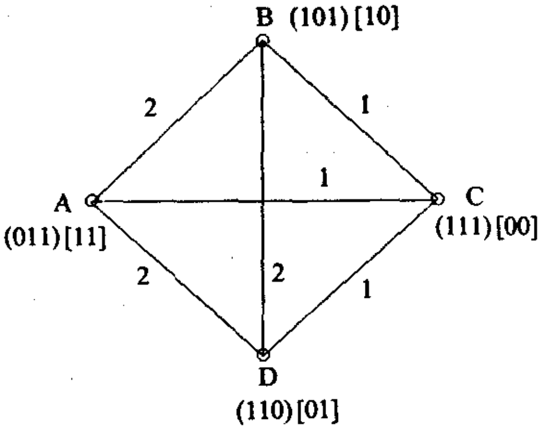


Figure 2. Graph for the four test vectors of the circuit in Fig. 1.

As per *rule* 2, one obtains the following 12 vector sets for PVS:

1	{11, 10, 01, 00}
2	{11, 01, 10, 00}
3	{10, 11, 01, 00}
4	{10, 01, 11, 00}
5	{00, 10, 11, 01}
6	{00, 10, 01, 11}
7	{00, 11, 01, 10}
8	{00, 11, 10, 01}
9	{00, 01, 10, 11}
10	{00, 01, 11, 10}
11	{01, 11, 10, 00}
12	{01, 10, 11, 00}

These 12 vector sets will have the maximum activity, and hence, maximum toggle coverage. Any of these test vector sets will be sufficient for the application.

2.3 Toggle Coverage Analysis

The toggle coverage analysis is given in Table 1.

Table 1. Vector ordering

Case No.	Vector order			
Case 1	00	10	11	01
Case 2	00	10	01	11
Case 3	00	01	10	11
Case 4	00	01	11	10
Case 5	00	11	01	10
Case 6	00	11	10	01
Case 7	11	10	00	01
Case 8	11	10	01	00
Case 9	11	01	10	00
Case 10	11	01	00	10
Case 11	11	00	01	10
Case 12	11	00	10	01
Case 13	10	11	00	01
Case 14	10	11	01	00
Case 15	10	01	11	00
Case 16	10	01	00	11
Case 17	10	00	01	11
Case 18	10	00	11	01
Case 19	01	11	00	10
Case 20	01	11	10	00
Case 21	01	10	11	00
Case 22	01	10	00	11
Case 23	01	00	10	11
Case 24	01	00	11	10

For the four vectors, maximum 4! = 24 re-ordering cases are possible. All the 24 cases and the respective toggle coverage have been listed (Table 2).

Table 2. Toggle coverage

Vector set	Toggling of each node			Total toggling	Total toggle coverage
	P	Q	R		(%)
Case 1	2	2	1	5	83.3
Case 2	1	2	2	5	83.3
Case 3	1	2	2	5	83.3
Case 4	2	1	2	5	83.3
Case 5	2	1	2	5	83.3
Case 6	2	2	1	5	83.3
Case 7	1	2	1	4	66.7
Case 8	1	2	2	5	83.3
Case 9	1	2	2	5	83.3
Case 10	1	1	2	4	66.7
Case 11	1	1	2	4	66.7
Case 12	1	2	1	4	66.7
Case 13	2	1	1	4	66.7
Case 14	2	1	2	5	83.3
Case 15	2	1	2	5	83.3
Case 16	1	1	2	4	66.7
Case 17	1	1	2	4	66.7
Case 18	2	1	1	4	66.7
Case 19	2	1	1	4	66.7
Case 20	2	2	1	5	83.3
Case 21	2	2	1	5	83.3
Case 22	1	2	1	4	66.7
Case 23	1	2	1	4	66.7
Case 24	2	1	1	4	66.7

It has been observed that ordering by this method results in the highest toggle coverage sequence.

3. RESULTS

The above algorithm has been exercised on four different circuits whose characteristics are given in Table 3. The toggle coverage of the circuit has been computed. Table 3 lists the circuit and its toggle coverage for power vectors.

Table 3. Circuit property and toggle coverage

Circuit No.	Circuit property	Vector length	Toggle coverage (%)
1	4 nand - reconvergent	4	83.3
2	nand-and-or	4	100.0
3	3 or-nor-and	5	100.0
4	4 nand - reconvergent (PI = 4)	5	100.0

4. CONCLUSION

A method for generating and ordering of test vectors has been presented to achieve maximum circuit activity. The method generates test vectors from a gate-level description of the circuit. These vectors can be used as burn-in test vectors for VLSI devices by targeting for maximum power dissipation. This algorithm has been exercised on different circuits and the toggle coverage computed. It has been shown that ordering by this method results in the generation of the highest toggle coverage sequence. The method is robust enough to handle various gate configurations.

ACKNOWLEDGEMENTS

The authors gratefully acknowledge Shri R. Ganesan, Head, QDTE, Vikram Sarabhai Space Centre, Thiruvananthapuram for his encouragement during various phases of realisation of this work. They are also grateful to Shri C.A. Ignatious, Head, Screening Laboratory for his constant incitement and guidance in formulating this work; to Smt E. Sujatha, FCD and Shri D. Basu, Group Director, ELG for their constructive suggestions in formulating this methodology for the testing of CMOS VLSI circuits.

REFERENCES

1. Rabaey, Jan M. Digital integrated circuits–a design perspective. Prentice Hall of India Pvt. Ltd., 2000.

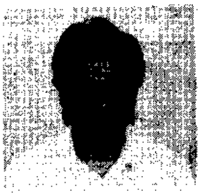
2. Tharakan, K.T. Oommen; Jacob, James & Srinivas, M.K. An efficient rule-based fault simulator. *In* VLSI Design'92, Proceedings of the 5th International Conference on VLSI Design, January 1992. pp.154–56.

3. Abraham, J. & Fuchs, W. Fault and error models for VLSI. *Proceedings IEEE*, May 1986, 639–54.

4. Campbell, M. Monitored burn-in: A case study on *in-situ* testing and reliability studies, *In* International Test Conference, IEEE, 1984. pp. 518–23.

Contributors

Mr KT Oommen Tharakan received his BTech (Electronics and Communication Engg) from the University of Kerala and ME (Electrical and Communication Engg) from the Indian Institute of Science, Bangalore. He is working at the Components Screening Laboratory, QDTE of Vikram Sarabhai Space Centre, Thiruvananthapuram. He is a Doctoral student at the Indian Institute of Technology Bombay, Mumbai. He has published a number of papers in the national/international journals. His research interests include: VLSI testing, fault modelling and test generation of FPGAs, and formal verification of very high speed integrated circuit hardware description language (VHDL) designs.



Dr SSSP Rao received his MTech (Applied Electronics) and PhD (Computer Science) from the Indian Institute of Technology (IIT) Bombay, Mumbai. He has been a faculty member in the Department of Computer Science and Engineering at the Indian Institute of Technology Bombay, Mumbai, since 1966 and working as Professor since 1985. At present, he is also Incharge, VLSI Design Centre, IIT Bombay. He is consultant to Indian Industries, viz., Tata Infotech Ltd, Mumbai, Mahyco and advisor to Switch-on Networks, USA; Seeyes Network Technologies. He has contributed 47 papers in national/international journals. He is also a member of various professional bodies, such as IEEE Computer Society, ACM, VLSI Society of India, CSI, ISTE, IETE (fellow). Besides, he is also a member of Global FPGA Technical Advisory Board of ST Microelectronics. His areas of interest are: VLSI design, advanced microprocessor system, advanced computer architecture, and reconfigurable computing.