Robust Test Generation for Power Supply Noise induced Path Delay Faults*

Xiang Fu[1,2], Huawei Li[1], Yu Hu[1], Xiaowei Li[1]
1 Key Laboratory of Computer System and Architecture, Institute of Computing Technology, Chinese Academy of Sciences
2 Graduate School of Chinese Academy of Sciences, Beijing, China
{fuxiang, lihuawei, huyu, lxw}@ict.ac.cn

***Abstract*-In deep sub-micron designs, the delay caused by power supply noise (PSN) can no longer be ignored. A PSN-induced path delay fault (PSNPDF) model is proposed in this paper, and should be tested to enhance chip quality. Based on precise timing analysis, we also propose a robust test generation technique for PSNPDF. Concept of timing window is introduced into the PSNPDF model. If two devices in the same feed region simultaneously switch in the same direction, the current waveform of the two devices will have an overlap and excessive PSN will be produced. Experimental results on ISCAS'89 circuits showed test generation can be finished in a few seconds.**

I. INTRODUCTION

Semiconductor industry has been driven by Moore's law for many years. As the dimension of device size continues to shrink, high-end processors are approaching to higher operating frequency and lower power supply voltage, which makes the performance of chips more sensitive to power supply noise (PSN). The sensitivity rises as the power supply voltage declines [1].

PSN is caused by excessively simultaneous switching of devices. Generally, the more the devices switch in the circuit, the worse the PSN will be. Since the simultaneous switching ratio of devices is determined by the stimulus applied to the circuit, PSN is highly pattern dependent. In designers' perspective, generating patterns that can induce worst case power supply noise is very important. During the design phase, the noise margin, in other words, the performance impact of PSN, should be estimated using appropriate patterns. While during the testing phase, a chip should be fully tested under conditions of excessive PSN.

Several researches have shed light on the PSN effect. [1] presented approaches based on a statistical analysis model, and it has been proved to be efficient in analyzing noise effect. In [2], a modeling approach, known as droop fault, is introduced. This model represents the phenomenon that, for a certain device, the simultaneous switching of its vicinities will result in slow-to-rise or slow-to-fall transition effect. If this device has a very tight slack, this effect will lead to functional failure of the circuit. [3] and [4] concentrate on low power test pattern generation in order to avoid over testing. [5], [6] and [7] focus on automatic test pattern generation (ATPG) techniques considering PSN effects.

* Corresponding author: Huawei Li.
This paper is supported in part by National Natural Science Foundation of China (NSFC) under grant No. 60776031, 60606008, 60633060, and in part by National Basic Research Program of China (973) under grant No. 2005CB321605.

In this paper, we target generating robust delay test patterns that could induce excessive PSN. *Timing window* is used to deal with timing information. Technically, PSN is a local phenomenon that can be analyzed in the layout of a chip. In order to excite the maximum PSN, each gate on a target path and its neighboring devices should be switched almost at the same time so that the switch current induced by simultaneous switching can reach its uppermost. Since not all these devices can be switched simultaneously, a good ATPG strategy is to launch transitions at the devices in the same timing window as much as possible. For this purpose, we incorporate static timing analysis (STA) into ATPG to identify the timing window that each device belongs to. The paths under test (PUT) are selected with a slack close to zero. Each PUT is treated as a victim path. The aggressor list of a victim path is generated depending on the results of static timing analysis and the layout information.

In the proposed ATPG algorithm, a 10-valued logic is used. Experimental results on ISCAS'89 benchmark circuits showed that the proposed method works efficiently for circuits of reasonable sizes, and the CPU time is acceptable.

The rest of the paper is organized as follows. Section 2 provides backgrounds on power supply noise. Section 3 gives a brief description of the fault model and the path sensitization criterion. Section 4 describes the aggressor list generation and ATPG flow. Experimental results on ISCAS'89 benchmark circuits are showed in Section 5. Finally, we present our conclusions in Section 6.

II. BACKGROUND

The path-delay fault model considers the cumulative effect of the delays along a specific combinational path in the circuit. When noise effects are considered, the delay of a path becomes uncertain. So it's necessary to generate test patterns that have taken noise effect into account as an enhancement of conventional delay test.

PSN consists of two parts: inductive ΔI noise and power net resistance induced voltage variation. The inductive ΔI noise is caused by the change of instantaneous current on either package lead inductance or wire/substrate inductance. The voltage variation of this part is in proportion to $L * di/dt$. The power net resistance induced voltage variation is caused by the current though resistive power and ground stripes, which further consists of two parts: one is power supply IR drop, the other is ground bounce.

The on-chip power grid is characterized as show in Figure 1. Power and ground stripes are connected to devices by

978-1-4244-1922-7/08/$25.00 ©2008 IEEE

M2/M3 vias. When transitions are launched, current is induced by the switching device.

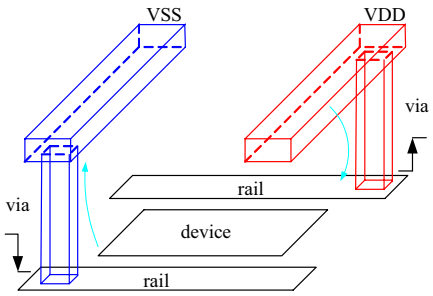

Figure 1. Characterization of power grid

The direction of the switching current is determined by the type of transition (low-to-high or high-to-low) launched at the device. When a low-to-high transition is launched at the inverter, the capacitor is charged and draws current form power stripes. As the resistance of the stripes cannot be ignored, the real voltage arrives at the device is a bit lower than the power supply voltage. We call this effect as power supply voltage drop. On the other hand, when a high-to-low transition is launched at the inverter, the load capacitor discharges, and the ground voltage becomes higher than zero. We call this effect as ground bounce. A timing model is presented in [8] to analyze power supply voltage drop and ground bounce effect.

The power grid can be modeled as a RLC network. In order to simplify the problem, [9] considers the power grid in steady-state case, and it can be solved by DC analysis. The power sources are modeled as simple constant voltage sources and power drains are modeled as independent current sources. As a result, the power grid problem can be transferred to a large linear system problem and the voltage of each node can be calculated.

III. Fault Model and Path Sensitization

Since conventional delay test model does not take PSN effect into consideration, a delay test, in which PSN-induced delay effects may not be activated, is insufficient. We propose a new fault model to address this issue.

3.1 PSN-induced Path Delay Fault (PSNPDF)

The physical implementation of a single path is illustrated in Figure 2. The feed region of a target via is defined as a list of logic blocks that contribute current to that via. A critical path may consist of gates that belong to different feed regions. Each gate of the path is influenced by the activity of the logic block in the same feed region. The voltage variation of gates in the same feed region can be reflected by the peak current consumed by simultaneous switching. The goal of test generation is to find vectors maximizing the switching current in the feed regions. As both of power supply voltage drop and ground bounce have delay impact on path delay, we make no difference between the two effects and only consider the maximum transition current.

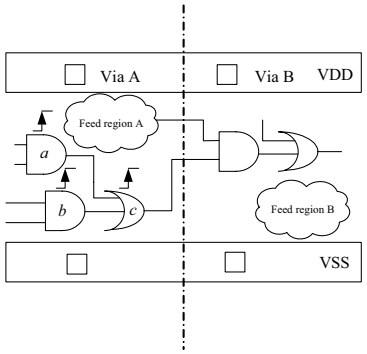

Figure 2. Target via and its feed region

Based on normally used aggressor/victim model, we treat each critical path as a victim, while the aggressor list of the path is generated based on static timing analysis. As shown in Figure 2, the gates *a*, *b* and *c* belong to the same feed region. Their switching currents absorbed from the corresponding vias are *Ia*, *Ib* and *Ic*. If *a*, *b* and *c* transit at the same time in the same direction, the peak current can reach its uttermost.

As it is quite different for a device to propagate rising transitions and falling transitions, a physical path has two logical paths: *lp0* and *lp1*. Each logical path corresponds to a path delay fault. A PSN-induced path delay fault (PSNPDF) is regarded as a victim critical path combined with an aggressor sub-path, the end of which is both in the timing window and the feed region of at least one gate on the critical path. In this fault model, precise static timing analysis is used to determine timing window that each device belongs to. The width of a timing window is two times of the mean transition time of the devices.

As usually referred, we use *on-path gates* and *on-inputs* to denote gates and lines on the PUT, while *side-inputs* are inputs of the on-path gates that are not on the PUT.

An PSNPDF of a logic path *LP* can be represented as *F(LP, sp-a)*. *LP* stands for a critical path. *sp-a* is the aggressor sub-path that satisfies the following conditions:

1) *LP* must be a robustly testable path.

2) *sp-a* is represented in the form of *sp-a(G-a,G-v)*. The sub-path, *sp-a(G-a,G-v)*, ends at gate *G-a*. *G-v* is an on-path gate of *LP* and belongs to the same feed region as that of *G-a*. *G-a/G-v* is normally considered as an aggressor/victim pair.

3) Suppose the delay of the sub-path ends at *G-a* in *sp-a(G-a,G-v)* is *d(sp-a)* and the delay of the sub-path (a sequence of gates on *LP*) to *G-v* is *d(sp-v)*. Given a specified small time interval δ to indicate that the signal change on *G-v* and *G-a* must occur within δ time interval, we have: $|d(sp\text{-}v) - d(sp\text{-}a)| \le \delta$. Under this condition, the current waveform of the two gates will have an overlap in the time interval δ.

δ is the width of the timing window and can be specified according to different technologies. Here, δ is two times of the mean transition time of devices. It is noted that there may

be several logical sub-paths ending at *G-a* that satisfy the above conditions, but one test that could sensitize one of the sub-paths is enough to activate the PSNPDF. As a result, test generation will stop when the first test is found for one of the sub-paths. If no test is found for all the sub-paths represented by *sp-a(G-a,G-v)*, the PSNPDF *F(LP, sp-a)* is an untestable fault.

3.2 Path sensitization criterion

For a multi-input gate, it has two kinds of non-controlling values: to-non-controlling transition and static non-controlling value. During delay test generation, the value of the side-inputs of a gate are usually restricted to non-controlling values implied by the two vectors <V1, V2>, this is called path sensitization criterion. There are three classes of path sensitization criterion commonly used in delay test generation: functional sensitization, robust test and non-robust test.

Path sensitization criterion for robust test generation is as follows.

1. To propagate a to-controlling transition at an on-input of an on-path gate, <V1, V2> should imply static non-controlling value at all side inputs of the gate.
2. To propagate a to-non-controlling transition at an on-input of an on-path gate, V2 should imply non-controlling value at all side-inputs of the gate.

In the ATPG procedure a 10-value logic is used. The path sensitization criterion of robust test generation is applied to the PUT of a target PSNPDF, while path sensitization criterion for sub-paths of the PSNPDF is relaxed to that of functional sensitization in order to increase the possibility to generate aggressor transitions during test generation [10].

IV. Aggressor List Generation and ATPG

Before we generate a test pattern for a PSNPDF *F(LP, sp-a)*, the PUT *LP* must be sensitized firstly. This will determine whether *LP* is robustly testable. The path sensitization criterion confines the side-inputs of the PUT to certain values. Thus, for an aggressor *sp-a(G-a,G-v)*, the transition direction of *G-a* and *G-v* may be pre-determined by path sensitization procedure. So we generate the aggressor list after the PUT is sensitized. This section will introduce the aggressor list generation and ATPG procedure.

The aggressor list of a victim path is actually a set of logical sub-paths. We adopt path identifier [11-12] to represent a path which is an efficient numeric representation of a path. It gives each path a unique integer identifier indicating its topological position.

For a PSNPDF *F(LP,sp-a)* of *LP*, the *sp-a(G-a,G-v)* stands for a logical sub-paths: a sub-path ending at *G-a* which is in the same timing window with *G-v*.

The aggressor list of of *LP* is denoted as SP-A and it is generated in the following steps:

1) For each on-path gate *G-v* of *LP*, we first find the all of the gates in the same feed region with *G-v*. These gates are potential aggressor gates of *LP* and each is denoted as *G-a*. Then, the path identifier is used to calculate the number of sub-paths going through these gates. After this step, we will get a sub-path set SP-Init. The elements in this set are potential aggressor sub-paths of *LP* and is denoted as *sp-init(G-a,G-v)*.

2) Static timing analysis is used to calculate the delay of sub-paths in SP-Init.

3) For a sub-path *sp-init(G-a,G-v)* in set SP-Init, if its delay is in the same timing window as that of *G-v*, we mark the path and reserve it in a new set SP-STA. Otherwise it is excluded from SP-STA. After this step, the elements in this SP-STA set are candidate aggressors of *LP*, and are represented by *sp-sta(G-a,G-v)*.

4) As *LP* is sensitized before aggressor generation procedure, certain logical sub-paths can be further excluded from SP-STA based on the current state that sensitizes *LP*. After this step, we will get the aggressor list SP-A. The elements in SP-A is denoted as *sp-a(G-a,G-v)*.

5) All the sub-paths of LP are sorted in the order of peak transition current of *G-a*. An aggressor gate whose peak transition current is higher will be given higher priority in the sub-path sensitization process.

The aggressor generation strategy can be illustrated in figure 3. The critical path *lp0* is high lighted in red. In *feed region1*, the SP-Init set is {(*a*, *c*), (*a*, *d*), (*b*, *c*), (*b*, *d*)}. Static timing analysis shows that gate *a*, *c* and *d* are in the same timing window. As a result, the set SP-STA is {(*a*, *c*), (*a*, *d*)}. In path sensitization phase, the side-input of gate *b* should be static non-controlling value *'0'* and the side-input of gate *e* should be a to-non-controlling value. Thus the set SP-A is {(*a*, *d*)} and the target fault is *f(lp0, (a,d))*.

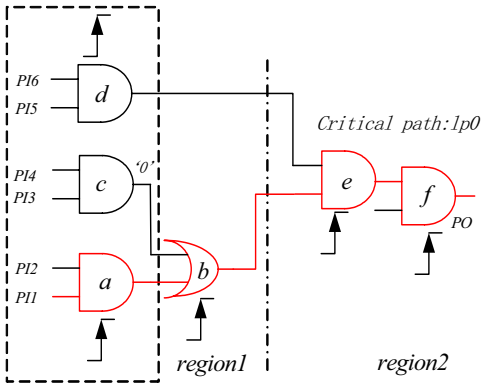

Figure 3. An example circuit for aggressor list generation

The ATPG flow is illustrated in Figure 4. The netlist is firstly processed to build the data structure we need. Then static timing analysis is used to generate timing information and the critical path set. After the PUT is sensitized, the timing and layout information is used in aggressor list generation procedure. Finally, the sub-paths in the aggressor list are sensitized and the test pattern is generated. Based on

the generated pattern, a fault simulation procedure is executed to improve ATPG efficiency.

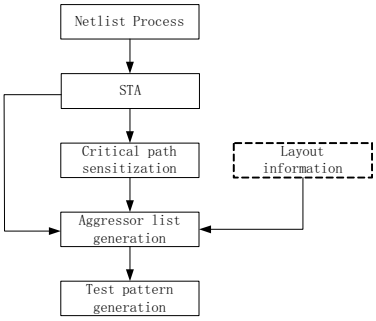

Figure 4. ATPG flow of Robust test generation considering PSN

V. EXPERIMENTAL RESULT

In this paper, a fanout weighted delay model is used to assign delay information. Due to lack of layout information, the devices of a feed region are selected randomly. However, we reserved interfaces both for other delay models and layout information in our program.

Experimental results on ISCAS'89 benchmark circuits are shown in Table 1. Column 1 shows the names of the circuits. Column 2 shows the numbers of selected target PSNPDFs collected after aggressor list generation procedure. The information about testable faults are shown in column 3, where the numbers of testable faults (#TFs) and the numbers of untestable faults (#UTF) are shown in its first two sub-columns respectively, and the numbers of aborted faults (#AFs) are shown in sub-column 3. The test efficiency, which equals to (#TFs+#UTFs)/#Fs, is shown in column 4. The fault coverages (FC) of these circuits are shown in column 5. The last column shows the CPU time in seconds. The CPU time includes time of target fault selection of CUT and robust test pattern generation for the target faults.

Table 1. Experimental results on ISCAS'89 circuits

Circuit Names	# of target faults	Testable Faults			Fault Efficiency (%)	Fault coverage (%)	CPU time (secs)
		#TFs	#UTF	#AFs			
s208	79	69	10	0	100.00	87.34	0.187
s386	454	219	235	0	100.00	48.24	0.296
s526	22	20	2	0	100.00	90.91	0.171
s820	187	101	82	4	97.86	54.01	0.281
s953	156	132	24	0	100.00	84.62	0.343
s1196	127	55	72	0	100.00	43.31	0.281
s1488	68	34	34	0	100.00	50.00	0.265
s1494	62	31	31	0	100.00	50.00	0.265
s5378	3103	2965	123	6	99.80	95.55	5.546

For larger ISCAS'89 benchmark circuits, most of the critical paths are robustly untestable. As a result, non-robust test patterns should be generated for these circuits and our future work will tackle this problem. As it is shown in Table 1 the CPU run time is determined by both of the circuits' scale and the target fault number. The fault efficiency is high and fault coverage acceptable. Since the number of aggressor/victim pairs is not large, the test patterns can be generated in seconds.

VI. CONCLUSIONS

In this paper, a new fault model: power supply noise-induced path delay fault (PSNPDF) model is proposed. Then, a robust test generation technique is presented based on this fault model. The path sensitization criterion of robust test generation is applied to the PUT of a target PSNPDF, while path sensitization criterion for aggressor sub-paths of the PSNPDF is relaxed to that of functional sensitization in order to increase the possibility to generate aggressor transitions during test generation. Experimental results on ISCAS'89 circuits showed test generation can be finished in a few seconds.

VII. REFERENCES

[1] J. Liou; A. Krstic, Y. Jiang, K.-T. Cheng, "Path selection and pattern generation for dynamic timing analysis considering power supply noise effects" *proc. ICCAD 2000* , pp. 493 – 496.

[2] C. Tirumurti, S. Kundu, S. Sur-Kolay, Yi-Shing Chang, "Modeling Approach for Addressing Power Supply Switching Noise Related Failures of Integrated Circuits", *Proc. DATE 2004* pp.1078 – 1083.

[3] J. Wang, Z. Yue, X. Lu, W. Qiu, W. Shi, D.M.H. Walker, "A vector-based approach for power supply noise analysis in test compaction", *International Test Conference 2005*.

[4] X. Wen, Y. Yamashita, S. Morishima, S. Kajihara, L.-T. Wang; K.K. Saluja, K. Kinoshita, "Low-capture-power test generation for scan-based at-speed testing" *International Test Conference 2005*, pp. 1019 – 1028.

[5] D. Mitra, S. Bhattacharjee, S. Sur-Kolay, B.B. Bhattacharya, S.T. Zachariah, S. Kunduc, "Test pattern generation for power supply droop faults", *Proc. the 19th International Conference on VLSI Design 2006*, pp. 343 – 348.

[6] A.Krstic, Y. Jiang, K. Cheng, "Delay testing considering power supply noise effects" *International Test Conference1999*, pp. 181 – 190.

[7] M. Nourani, A. Radhakrishnan, "Power-supply noise in SoCs: ATPG, estimation and control" *International Test Conference 2005*.

[8] M. Shao, Y. Gao, L. Yuan, M.D.R. Wong, "IR drop and ground bounce awareness timing model" *Proc. IEEE Computer Society Annual Symposium on VLSI 2005*, pp. 226 – 231.

[9] Y. Zhong, M.D.F. Wong, "Fast algorithms for IR drop analysis in large power grid" *proc. ICCAD 2005*, pp. 351 – 357.

[10] H. Li, P. Shen; X. Li; "Robust test generation for precise crosstalk-induced path delay fault", *Proc. VLSI Test Symposium 2006*. pp. 300 – 305.

[11] I. Pomeranz, L. N. Reddy, and S. M. Reddy, "SPADES: A Simulator for Path Delay Faults in Sequential Circuits", *Proceeding of European Conf. on Design Automation 1992*, pp. 428-432.

[12] W. Long, Z. Li, S. Yang, and Y. Min, "Memory Efficient ATPG for Path Delay Faults," *Proceedings of Sixth Asain Test Symposium 1997*, pp. 326-331.