Power Droop Testing

Ilia Polian[1], Alejandro Czutro[1], Sandip Kundu[2] and Bernd Becker[1]

[1]Albert-Ludwigs-University
Georges-Köhler-Allee 51
79110 Freiburg, Germany
{polian|aczutro|becker}@informatik.uni-freiburg.de

[2]Department of ECE
University of Massachusetts
Amherst, MA 01003, USA
kundu@ecs.umass.edu

Abstract— **Circuit activity is a function of input patterns. When circuit activity changes abruptly, it can cause sudden drop or rise in power supply voltage. This change is known as power droop and is an instance of power supply noise. Although power droop may cause an IC to fail, such failures cannot currently be screened during testing as it is not covered by conventional fault models. In this paper we present a technique for screening such failures. We propose a heuristic method to generate test sequences which create worst-case power drop by accumulating the high-frequency and low-frequency effects. The generated patterns need to be sequential even for scan designs. We employ a *dynamically constrained* version of the classical D-algorithm for test generation, i.e., the algorithm generates new constraints on-the-fly depending on previous assignments. The obtained patterns can be used for manufacturing testing as well as for early silicon validation. A prototype ATPG is implemented to demonstrate the feasibility of the approach and test sequences are generated for ISCAS circuits.**

Index Terms— **Power droop, Signal integrity errors, ATPG**

I. Introduction

State-of-the-art high-performance digital ICs manufactured in deep-submicron technologies tend to draw considerable amounts of power during operation. Large transients, i.e., sharp changes in power consumption, are possible within a few clock cycles. One example is a microprocessor in the idle mode which has to start complex calculations that involve using multiple fixed-point units and floating-point units simultaneously. The difference between power consumption in idle mode and peak power consumption may exceed 100 watts. The transition may take around one nanosecond on a multi-GHz machine.

Power droop describes the impact of power consumption transients on logic values on the signal lines of a circuit and ultimately the correctness of its operation [1], [2]. It is related, yet not identical, to static IR drop and ground bounce. We distinguish between low-frequency and high-frequency power droop. Low-frequency power droop occurs when the VRM (voltage regulator module) is unable to handle large transients in the power consumption of the whole device due to non-negligible inductance of the interconnect. Not enough current is provided by the VRM during a very short period of time, resulting in a drop of voltage on certain logic lines (power starvation). Because of capacitive effects described in detail in Section II-A it takes a number of clock cycles until low-frequency power droop creates the largest impact. This problem is known in the automatic test equipment design community [3].

High-frequency power droop also creates power starvation, however its mechanism is different. The reason for high-frequency power droop, which is closely related to ground bounce and simultaneous switching noise (SSN) [4], is the limited capability of the power distribution network on-chip to deliver power quickly enough due to, e.g., insufficient sizing of power rails or vias. When there is a logic value transition on several lines supplied by the same segment of a power rail, the amount of power drawn by these simultaneous transitions might exceed the amount of power which the power grid can deliver in the given short time. As a result, the switching time will increase, potentially leading to a delay fault.

In this paper we propose an automatic test pattern generation (ATPG) algorithm which attempts to create worst-case power droop conditions by combining the effects of low-frequency and high-frequency power droop. The generated sequence can be used for evaluating early silicon for design flaws such as an insufficient sizing of the power grid, in particular vias. This information may be difficult to obtain analytically before actually manufacturing the IC. The second application of the generated test sequence is in manufacturing test. Since power droop belongs to the class of *circuit marginalities* [5], some ICs may be affected stronger than others. By applying the generated sequence, the ICs which are vulnerable to power droop can be identified and either rejected or binned as low-performance parts. The patterns can be used in presence or absence of special on-die droop detectors [2].

The proposed ATPG procedure addresses low-frequency power droop by first generating a sub-sequence which leads to low switching activity (and thus power consumption) in the circuit under test, followed by a sub-sequence which maximizes global switching activity. As will be explained in the next section, the voltage levels in the circuit will be dropping for several clock cycles during the application of the second subsequence before they reach a minimum and start rising again. In this clock cycle, worst-case high-frequency power droop is created by imposing simultaneous transitions in the same direction on a victim line and on several aggressor lines which are all provided with power by the same segment of the power grid. If the circuit is vulnerable to power droop, a delay fault results on the victim line, which is propagated to an observable point. The need for sub-sequences with low and high switching activity prohibits using scan for any test

1-4244-9707-X/06/$20.00 ©2006 IEEE

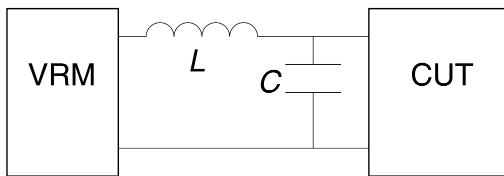

Fig. 1. Circuit under test (CUT) connected to voltage regulator module (VRM), including capacitor C and parasitic inductance of interconnect L

vector of the sequence except the very first one. As a result, sequential test generation is required. The algorithm proposed in this paper is based on constrained sequential test generation. Constraints are employed to maximizing low-frequency and high-frequency power droop; some constraints are added to the ATPG instance during the execution of the algorithm.

Signal integrity testing is often associated with effects between logic lines, including capacitive crosstalk [6], [7], [8], [9], inductive oscillatory noise [10] and their combination [11]. However, power grid induced signal integrity issues recently received some attention. Ground bounce was targeted in [6], [12], [13]. In [14], noise is maximized on gates belonging to a sensitized path. The approach in [15] is based on library cell characterization. Lower-cost power supply noise estimation was discussed in [16], [17]. Silicon measurements using automated test equipment were used for test generation in [18]. Concurrent detection of power supply noise using on-chip monitors has been proposed in [19], [20]. In [21], [22], [23], test sequences which *avoid* overly high noise levels are generated. The motivation is that such noise levels do not show up in application and rejecting ICs based on tests with uncontrolled noise would result in overtesting and throwing away good parts. All of the mentioned papers considered noise imposed by one transition and did not model effects stretching over more than two cycles. Power droop is targeted specifically in [2] by means of an on-die detector. In [24], extensions of test access mechanisms required for power droop testing are mentioned.

The remainder of the paper is organized as follows. The next section introduces the power droop phenomenon. Generation of a test sequence to provide evidence of power droop is discussed in Section III. The ATPG algorithm for solving the problem posed in Section III is described in Section IV. Experimental results are reported in Section V. Section VI concludes the paper.

II. Power Droop

In this section, the physics behind low-frequency and high-frequency power droop is presented in more detail.

A. Low-frequency power droop

Figure 1 shows the circuit under test (CUT) connected to power supply (VRM). The parasitic inductance of the interconnect is denoted by L. We call a sudden increase in current I demanded per unit time t (which is equivalent to a sudden increase in power consumption) a dI/dt event. After a dI/dt event, the CUT will see its power supply voltage V_{DD} reduced by $L \cdot dI/dt$. For a current transient of 100 amperes (which, for $V_{DD} \approx 1$ Volt, is equivalent to a power transient of 100 watts) taking place within 10^{-9} seconds or three cycles on a 3.3 GHz machine, this value is dramatic even for inductances L far below 1 nanohenry.

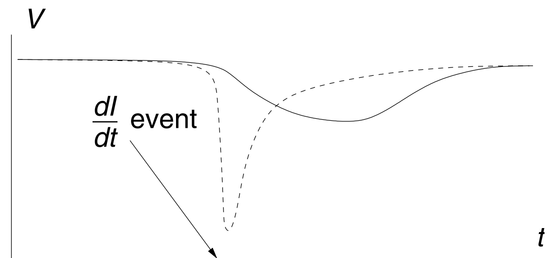

Fig. 2. Voltage seen by CUT after a dI/dT event, in absence and in presence of capacitor C

This effect is mitigated by adding a capacitance C as shown in Figure 1 to cover the CUT's short-term demand for current after a dI/dt event. The voltage droop per unit time induced by the load current is calculated as $dv/dt = I/C$ [3]. Since the inductance of the line between the capacitor and the CUT is much lower than L, the V_{DD} drop is much less severe. However, if C is discharged before the VRM is ready to supply the full amount of needed current, a V_{DD} drop is observed, albeit of a smaller extent and some time after the initial dI/dt event.

Figure 2 sketches the power supply voltage seen by the CUT after a dI/dt event. The dashed curve indicates the development over time in absence of a capacitor. V_{DD} falls down to almost 0 volt and then slowly recovers. The solid curve demonstrates the effect of the capacitor.[1] V_{DD} goes down much slower, and after some time a point is reached when the VRM provides enough current (and starts charging the capacitor again). Whether the V_{DD} drop is large enough to cause a logic or delay failure depends on the extent of the dI/dt event, i.e., on how much more current is demanded over how small a period of time, and the values of L and C as well as the characteristics of the VRM. It is important, however, that the impact is most severe several clock cycles after the actual dI/dt event.

B. High-frequency power droop

The power distribution network (power grid) of a state-of-the-art high-performance IC stretches over several metallization layers, connected by vias, as indicated by Figure 3. The topmost layers are often reserved for power rails and in clock distribution network while lower layers are shared with logic signal lines. In general, the power delivery capacity of a power rail is given by its width, which tends to decrease on lower layers. There is pressure to limit the width of power rails as the area consumed by them is not available to logic signal lines.

[1] The curve is for illustration only and has not been obtained by measurement or simulation. Refer to Figure 7 in [3] for a measured typical voltage response.

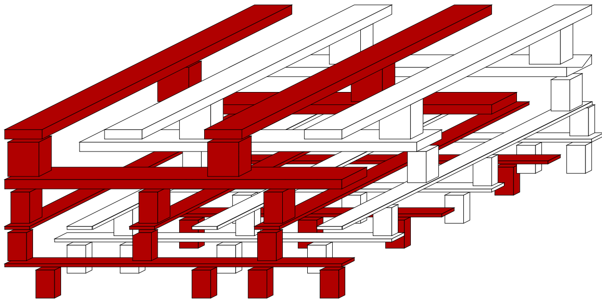

Fig. 3. 4-layer power grid (V_{DD} rails are shown in red, ground rails are shown in white)

The vias connecting power rails of different layers are relatively small and hence are an obvious bottleneck for power delivery. In addition, they are manufactured in the same dual-damascene technology as the vias connecting signal lines and consequently are prone to marginal defects such as lacking metal causing elevated resistance [25], [26]. The layer on which the logic cells are attached to the power grid can be as low as Metal 2, which means that there are six or seven metallization layers above and power must be delivered through a corresponding amount of vias. We call a part of power rail located between two vias a *segment*.

High-frequency power droop occurs when multiple cells drawing current from the same power grid segment suddenly increase their current demand. If the current cannot be provided quickly enough from other parts of the chip, power starvation results in a voltage drop. In contrast to low-frequency power droop, this is a highly localized and transient phenomenon: one of the involved cells is slowed down for one clock cycle. More details on the electrical modeling of high-frequency power droop can be found in [1], [2].

High-frequency power droop is very hard to debug or diagnose because it is nearly impossible to reproduce the error if the power droop conditions are not targeted specifically. Moreover, there is typically no hardware defect which could be identified using electrical or physical failure analysis [27]. The effects of power droop could be mistakenly classified as radiation-induced soft errors [28] because they are also unrepeatable, last for just one clock cycle, and no hardware defect is present. In contrast, identified high-frequency power droop points to either a design issue (inadequate power rail sizing) or a failed via of the power grid, i.e., a systematic manufacturing defect rather than random noise. It is thus important to target power droop systematically. As an extreme case, an erroneous assumption of increased levels of radiation-induced soft errors could result in employment of mitigation techniques such as gate upsizing, which would increase the total power demand and eventually worsen the power droop.

III. Test Generation Problem

The goal of this work is to generate a test sequence which imposes worst possible power droop and exposes its presence. First, low-frequency power droop is induced by creating a global dI/dt event stretching over multiple cycles. When the voltage droop is most severe due to power starvation caused by the low-frequency power droop, high-frequency power droop is imposed on a *victim line* v. For this purpose, line v and *aggressor lines* a_1, a_2, ..., which are driven by logic cells drawing power from the same segment of the power grid as the cell driving line v, are required to switch simultaneously and in the same direction. Combined power starvation due to both low-frequency and high-frequency power droop leads to an increased switching delay on line v. Finally, a path from line v to an output or flip-flop is sensitized, such that the faulty effect can be observed.

To create a global dI/dt event, the amount of current drawn by the circuit from the VRM must change rapidly. This *global current demand* is a function of the inputs applied to the circuit. In deep-submicron CMOS, current is consumed for switching events of the gates and for leakage. In present-day manufacturing technology, switching current dominates leakage current. Leakage current may depend on the applied inputs because of second-order effects. However, the extent of pattern-dependent variation in leakage current is so small that we ignore it in this work and assume that the leakage current is a constant offset which cannot be influenced. In contrast, switching current is completely determined by the input sequence, as it is given by the (weighted) number of switching events in the circuit, i.e., nodes changing their logic value from 0 to 1 or vice versa.

The test sequence for imposing worst-case low-frequency power droop is composed of two subsequences $l_1 l_2 \dots l_M$ and $h_1 h_2 \dots h_N$. Subsequence $l_1 l_2 \dots l_M$ should minimize switching activity, i.e., the number of switching events, in the circuit, while subsequence $h_1 h_2 \dots h_N$ should maximize the switching activity (peak n-cycle power in the classification of [29]) in the circuit. In general, switching events on different lines consume different amount of power. This can be modeled by weighting the switching activity on a node, e.g., by the load it drives. Currently, we do not employ any such weighting, but it can be easily integrated into our framework. It is impossible in general that all the nodes in the circuit switch. For instance, if both inputs of an XOR gate have a switching event its output cannot have one.

Worst-case high-frequency power droop is given when the victim wire v and all the aggressor wires a_1, a_2, ... switch *in the same direction*. Then, a significant amount of current must be transported to or from a single segment of the power grid through a series of resistive and inductive vias. In general, it may not be possible to impose such transitions on all the aggressors simultaneously because of logic implications between them. For example, a_2 could be driven by an inverter fed by a_1. Hence, we require that *a possibly large number of aggressors* switch in the same direction as the victim. This is different from testing for capacitive crosstalk which requires *opposite* transitions on the aggressor nodes. High-frequency power droop leads to a delay fault on v which must be propagated to an observable point. Consequently, for testing the high-frequency power droop, we require a test pair (t_1, t_2), which detects the transition fault on v with additional constraints on aggressors. If an on-die droop monitor is available, the transition fault must only be activated but not propagated to an output [2].

The automatic test pattern generation (ATPG) problem for power droop is formalized now. We assume a full-scan sequential circuit. Extension to combinational circuits is straightforward. We assume that the transition on the victim line is rising, the problem formulation for falling transition is symmetric.

Problem: Power droop ATPG.
Input:

1) Circuit net-list on gate level.
2) Victim line v and list of aggressor lines a_1, a_2, ...for high-frequency power droop.
3) Length of low-switching-activity and high-switching-activity subsequences (LSS and HSS), M and N, respectively, for the low-frequency power droop.

Output: Initial state s_0 of the circuit and a sequence of $M+N$ input vectors $l_1 l_2 \dots l_{M-1} h_1 h_2 \dots h_{N-1} t_1 t_2$ for power droop detection, with the following constraints:

1) Line v has logic value 0 in time frame $M+N-1$, i.e., when vector t_1 is applied.
2) Stuck-at 0 fault at line v is detected in time frame $M+N$, i.e., under vector t_2 (if an on-die droop detector is used, no propagation to an output is needed [2]).
3) As many aggressor lines a_i as possible assume logic value 0 under vector t_1 and logic-1 under vector t_2.
4) Switching activity is as low as possible during the application of the first M vectors $l_1 l_2 \dots l_{M-1} h_1$.
5) Switching activity is as high as possible during the application of N vectors $h_1 h_2 \dots h_{N-1} t_1$. □

The obtained test sequence is applied as follows. First, the initial state is shifted into the flip-flops of the circuit using scan, and the input sequence is applied to its primary inputs. It is not possible to use scan within the sequence as this would violate the switching activity constraints 4 and 5. Finally, the primary output is read out and the state of the circuit is scanned out. Note that the low-switching-activity subsequence (LSS), the high-switching-activity subsequence (HSS) and the test pair (t_1, t_2) for high-frequency power droop are overlapping: vector h_1 belongs to both subsequences LSS and HSS and vector t_1 belongs to both HSS and the test pair.

Constraints 1 and 2 ensure that the delay imposed by power droop on the victim line (rising transition fault on v) is detected. Condition 3 creates worst-case high-frequency power droop. Conditions 4 and 5 help to induce the largest possible dI/dt event required for worst-case low-frequency power droop. Fault detection takes place when the combined effects of low-frequency and high-frequency power droop impact the delay on line v. While satisfaction of Constraints 1 and 2 is mandatory, the other constraints demand only satisfaction in as many cases as possible. These constraints might also be contradictory in sense that an assignment necessary to maintain high switching activity (Constraint 5) may prevent the values on aggressor lines desired by Constraint 3. Consequently, the problem may have multiple solutions with different degree of satisfaction of Constraints 3 through 5.

It may appear that the sequence generated by the power droop ATPG imposes conditions which are so unlikely to happen in normal operation that rejecting an IC based on this sequence results in overtesting. It cannot be ruled out that a given IC will never experience a significant dI/dt event in its lifetime. Still, the ability to process the generated sequence is part of IC's specification, so the customer is free to apply this sequence and to expect the IC to function correctly. The only possible source of overtesting is the initial state potentially being non-functional, i.e., not reachable from the initial state. There exist a variety of techniques to restrict the ATPG to functional initial states [30], [31], [32], [33], [34], [35], [36] which can be incorporated into the proposed methodology.

IV. ATPG Algorithm

The proposed method to generate a test sequence for power droop is based on the D-algorithm, which is a fundamental ATPG method [37], [38]. We modified the D-algorithm such as to consider Constraints 1 through 5 of the problem formulated in the last section. We describe the proposed algorithm first and then discuss a technique to improve its run time.

A. *Dynamically constrained D-algorithm*

The conventional D-algorithm generates a test vector for a stuck-at fault in a combinational circuit with controllable primary inputs and observable primary outputs. For a stuck-at 0 fault on line v, the D-algorithm searches for an input vector which (1) sets line v to logic 1 (*justification*) and (2) ensures that the faulty effect D on v is propagated to a primary output by sensitizing a path (*propagation*), where D means that v assumes value 1 in the fault free circuit and value 0 in the faulty circuit. This is done by assigning logic values to lines and calculating implications of that assignments. Whenever a value on a line is not implied by assignments made so far, a *decision* is made whether to set it to 0 or 1. This is continued until either a test vector detecting the fault is found or an inconsistency in assignments is identified. In the latter case, one of the earlier decisions is reversed and the search is continued with the opposite decision (*backtracking*). If no decisions are left to backtrack, then the fault is proven to be untestable.

The problem considered here is different from stuck-at test generation in combinational circuits, thus requiring modifications of the conventional D-algorithm. First, the circuit is sequential and a sequence of $M+N$ vectors is required for testing. This is handled by creating an $(M+N)$-time frame *unfolding* of the circuit. Primary inputs of every time frame are controllable. Since scan is assumed but can be operated only for setting the initial state and observing the final state, secondary inputs (flip-flops) are controllable only in the first time frame and observable only in the last time frame. Fault effect emerges in the last time frame, so it can be observed on a primary or a secondary output of that time frame. Unfolding is the standard approach for sequential test generation [38].

Second, additional constraints are required. While some of them, namely Constraints 1 and 2, are *mandatory*, Constraints 3 through 5 are *desired* to hold at as many lines as possible but cannot be expected to always hold. Additional mandatory constraints are straightforward to integrate into the D-algorithm; they have been used by several authors for test generation for non-stuck-at faults. Any assignment violating these constraints

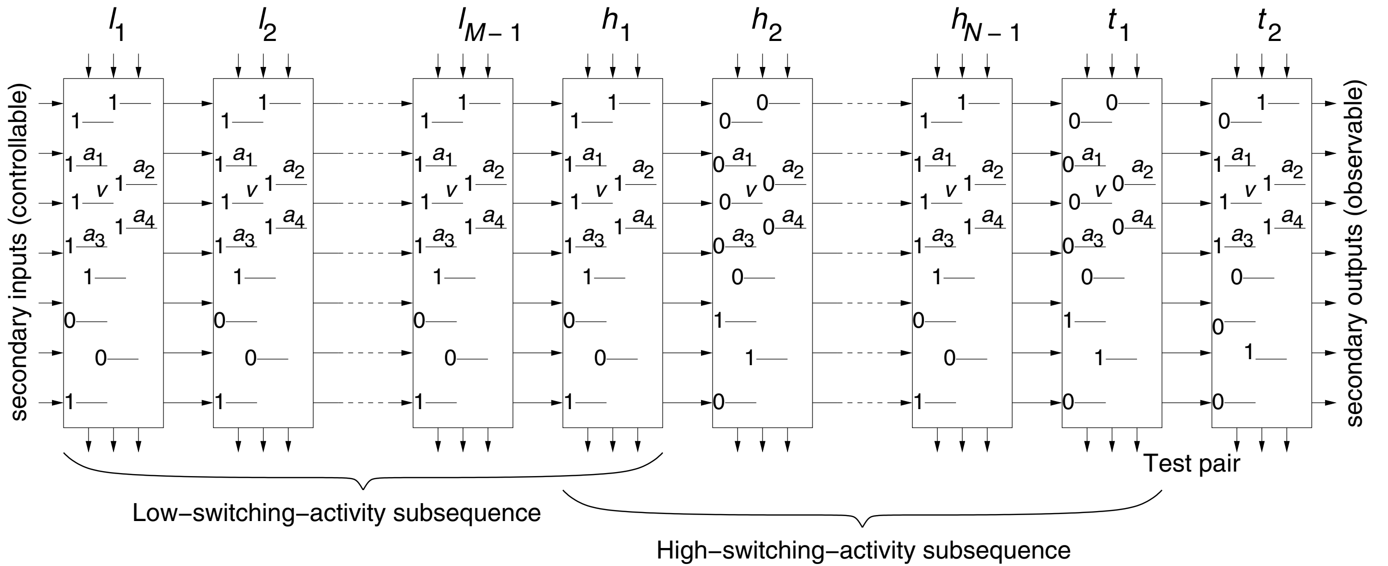

Fig. 4. Time frame expansion of the circuit for power droop ATPG

is not a valid solution and triggers a backtrack. In case of desired constraints, the maximal number of them should be satisfied simultaneously. Violating a desired constraints does not necessarily render the solution invalid. The handling of desired constraints is difficult because the achievable quality of a solution is not known. An assignment which is useful for satisfying a constraint may imply that other constraints elsewhere in the circuit are violated.

Figure 4 depicts a problem instance with all desired constraints satisfied. The victim is denoted v, four aggressors are denoted a_1 through a_4, and all other lines are not involved into high-frequency power droop. During LSS, no line is switching. During HSS, all lines are switching. When the test pair is applied, the victim and all the aggressors have a rising transition; the values on other lines are irrelevant.

In order to keep the problem tractable, we employ the following strategy. Mandatory constraints are enforced by adding them into the assignment queue of the D-algorithm. Regarding the desired constraints, the decision procedure of the D-algorithm is modified. Whenever a decision is to be made which either satisfies or violates a desired constraint, the satisfying assignment is tried first. Only if this assignment leads to an inconsistency will the D-algorithm try the violating assignment. In particular, we introduce the following three rules:

- Rule 1 Assume rising transition on the victim line v. When a decision is made on an aggressor line a_i in time frame $M+N-1$ (under vector t_1), logic 0 is always assigned first. In time frame $M+N$ (under vector t_2) logic 1 is always assigned first.
- Rule 2 When selecting which line to make a decision on, the lines in later time frames are always preferred.
- Rule 3 Suppose that the decision is made on line n in time frame k, $M \leq k \leq M+N-2$, i.e., under one of the vectors $h_1, \ldots, h_{N-1}$. If line n is already assigned in time frame $k+1$ to logic value $b \in \{0,1\}$, assign it to the opposite logic value $\bar{b}$. If the decision is made on line n in time frame k, $1 \leq k \leq M-1$, i.e., under one of the vectors $l_1, \ldots, l_{M-1}$, and line n is already assigned in time frame $k+1$ to logic value b, assign it to the same logic value b.

The rationale of Rule 1 is to create as many simultaneous transitions on aggressor lines to create worst-case high-frequency power droop. Rule 2's purpose is to facilitate the application of Rule 3. Rule 3 minimizes (maximizes) switching activity in LSS (HSS) to impose worst-case low-frequency power droop. If switching activity were weighted, Rule 2 could be modified such as to try the lines with highest weight first. Rule 3 is implemented by generation of desired constraints on-the-fly, i.e., whenever an assignment is done, a desired constraint for the preceding time frame is generated. We call the resulting algorithm *dynamically constrained D-algorithm.*

The proposed approach is heuristic, i.e., it may not yield the optimal result. Recall, however, that Constraints 3 through 5 may be conflicting and it is not clear what the optimal solution is anyway, as explained in the end of Section III. One part of power droop test generation is the creation of a sequence with worst-case switching activity. Even this sub-problem has been targeted but never solved optimally for non-trivial circuits [39], [40], [41], [42]. (Advanced search techniques for large solution spaces such as algebraic decision diagrams [43] and genetic algorithms [44], [29], [45] have been used instead).

B. Speeding up test generation

Scalability issues may require speed-up techniques to handling large circuits and large numbers of time frames to unroll, i.e., $M+N$. There are a variety of improved techniques in the basic ATPG algorithm and add-ons such as diverse learning strategies. One further option is to consider subsequences of the global test sequence $l_1 l_2 \ldots l_{M-1} h_1 h_2 \ldots h_{N-1} t_1 t_2$. For instance, let the sequence be $l_1 l_2 l_3 l_4 h_1 h_2 h_3 h_4 t_1 t_2$. One could first generate a sequence for $l_1 l_2 l_3$ and the initial state s_0

Circuit	# switching aggressors	Switching activity			Time, (CPU)
		LSS	HSS	Random	
c0017	3	0	87	47	0:00:00
c0095	3	0	71	39	0:00:00
c0880	3	0	50	34	0:00:01
c1908	4	0	57	40	0:00:01
c3540	3	0	46	33	0:00:05
c6288	3	0	47	38	0:00:39
c5315	4	0	53	40	0:00:14
c7552	5	0	52	41	0:00:42
s00027	2	11	69	37	0:00:00
s00208	1	0	36	19	0:00:00
s00298	1	10	50	36	0:00:01
s00386	2	4	30	36	0:00:02
s00382	2	4	32	34	0:00:02
s00344	3	16	47	35	0:00:02
s00349	1	3	40	35	0:00:01
s00400	2	5	37	34	0:00:02
s00444	1	0	39	31	0:00:07
s00526	1	3	24	28	0:00:01
s00510	1	0	29	24	0:00:04
s00420	2	3	26	13	0:00:02
s00832	3	0	37	30	0:00:04
s00820	2	0	43	30	0:00:01
s635	1	0	13	10	0:00:02
s00641	4	0	50	31	0:00:06
s00953	1	1	19	17	0:05:06
s00713	2	2	49	29	0:00:15
s00838	3	0	27	10	0:00:04
s938	1	0	28	10	0:00:05
s01238	1	13	36	26	0:00:23
s01196	3	18	38	27	0:00:21
s01494	1	29	34	31	0:00:32
s01488	1	9	33	31	0:00:56
s01423	1	5	26	32	0:00:46
s1512	1	0	29	24	0:00:15
s3271	5	32	55	49	0:00:55
s3384	4	23	58	46	0:01:39
s3330	3	9	52	36	0:01:06
s4863	3	24	40	42	1:34:17
s05378	2	6	53	35	0:08:38
s6669	4	18	45	38	0:06:25
s09234	4	16	41	29	0:35:08
Average	2.37	6.44	42.15	31.43	0:03:53

TABLE I

EXPERIMENTAL RESULTS FOR RISING TRANSITION AND $M = N = 10$, I.E., 20 TIME FRAMES

taking all the relevant constraints into account. Let the circuit state after three cycles be s_3. Then, test generation is run for subsequence $l_4h_1h_2$, with secondary inputs set to s_3 rather than being controllable. Let the resulting state be s_6. Finally, test generation is run for $h_3h_4t_1t_2$ with secondary inputs set to s_6. This technique reduces one ATPG run for a large number of time frames by several runs for a shorter numbers of time frames. In general, this approach could fail to generate a sequence which the ATPG proposed above, which considers the sequence as a whole, could produce, or the quality of the obtained sequence may be worse.

V. EXPERIMENTAL RESULTS

The problem under consideration takes the circuit net-list, the aggressor and the victim nodes, and the lengths of the low-switching-activity and high-switching-activity subsequences (LSS and HSS) as inputs. We applied the dynamically constrained D-algorithm for power droop test sequence generation to ISCAS 85 and 89 circuits. The determination of other problem inputs is generally out of the scope of this paper and requires layout, power grid and technology information not available for ISCAS circuits. The victim line v can be obtained by analysis in [1]. The aggressor lines are determined by the analysis of the power grid: the segment which supplies v's driver with current is identified. Aggressor lines are driven by cells powered by the same segment. M and N should be chosen such as to maximize the voltage drop (recall Figure 2). They can be derived analytically from the electrical parameters of the circuit, the VRM, the capacitor, etc., or obtained by measurement.

Instead, we selected the affected nodes randomly and ran the experiments for different lengths of LSS and HSS. We selected the node with the maximal fanout as the victim, as this node is likely to have the largest load. If there were several such nodes, we selected one node randomly. An interesting finding of [1] is that the number of possible sites for power droop is very limited: less than 100 for a microprocessor of 128,000 standard cells. We selected five random nodes in the circuit as aggressors. We repeated the experiment for aggressors selected based on their neighborship to the victim and obtained similar results. We generated results for both the rising and the falling transition on the victim node and different values of M and N.

Table I contains the results for circuits up to s9234 with length of LSS and HSS being equal to 10. The second column shows on how many of the five aggressors the supporting transition can be justified (Constraint 3). Columns 3 and 4 report the average switching activity (number of switching events per line and clock cycle) in per cent during LSS and HSS, respectively (Constraints 4 and 5). In order to put the generated numbers into a perspective, we determined the average switching activity of 5000 random vectors, which is reported in Column 5. Note that random vectors in [44] achieve much higher switching activity (often over 100%) as multiple switches within a cycle due to glitches are accounted for and a weighted metric is used. Execution time on a 2×1280 MHz UltraSPARC-IIIi machine with 6 GB RAM is shown in the last column. Average numbers are given in the last row of the table.

The quality of the resulting sequence is given by its ability to generate worst-case power droop. We are currently preparing a Spice simulation of a small circuit to determine the actual power droop. Alternatively, the generated patterns can be validated by applying commercial noise estimation tools. The worst-case high-frequency power droop is achieved when supporting transitions are justified on many of the aggressor nodes, five being the optimum. For approximately half of the considered circuits, the supporting transitions have been justified on three or more aggressors. The severity of the low-frequency power droop is given by the extent of the dI/dt event, i.e., the difference between switching activity during LSS and HSS. The switching activity during LSS is zero for combinational circuits and close to zero for 10 sequential

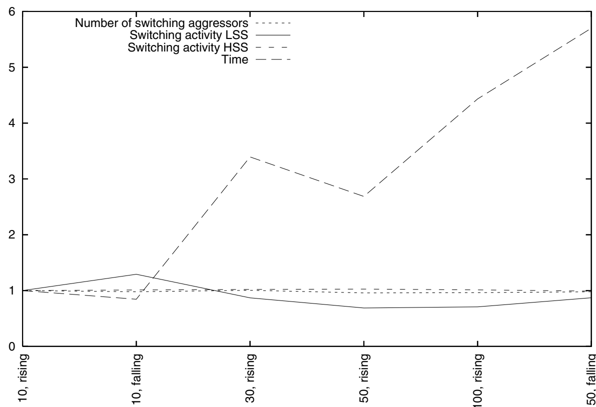

Fig. 5. Number of switching aggressors, switching activity and execution time for $M = N = 10$, 30, 50, 100 and 150 and rising and falling transitions (normalized)

circuits. For 13 out of remaining 23 circuits the switching activity during HSS exceeds its counterpart during LSS by factor 5 or more, while this ratio is less than 2 only for three circuits. Recall that 100% switching activity in HSS cannot be achieved in general and that the largest achievable value it is not even known.

Comparing switching activity during HSS to that of random vectors, the ratio is about 1.35 on average and exceeds 1.5 for 10 out of 41 circuits. Note that decisions previously made on LSS and the test pair may imply assignments on HSS which conflict with the goal of maximizing the switching activity. In fact, the switching activity during HSS falls below that of random vectors for five circuits due to such implications. Despite this fact, on average the generated HSS is superior to random vectors for which no such implications are required to be regarded.

We generated corresponding data for different values of M and N: 10, 30, 50, 100 and 150, and for rising and falling transitions on the victim line. We report only average numbers in graph form in Figure 5. The numbers are normalized to the average data for $M = N = 10$ and rising transition, reported in Table I. It can be seen that neither the number of switching aggressors nor the switching activity during HSS change much. The switching activity during LSS is improved (decreased) slightly for larger values of M and N. The solution quality seems to be relatively stable with respect to M and N and some of the statistical noise is attributed to the random sets of aggressors being not the same throughout the experiments.

Somewhat unexpected, the increase in execution time is sublinear in M and N. Moreover, this increase is not monotonic. The hardness of the problem seems to depend on the specific problem instance (controllability and observability of aggressor and victim lines etc.) rather than simply the number of unfolded frames given by $M + N$. The line assignments imposed by the desired constraints may not be helpful for finding a valid test sequence quickly, and this dependency seems to show up quite randomly.

Results for larger ISCAS 89 circuits are summarized in Table II. We did not generate the complete set of results for different values of M and N for these circuits. This experiment was run on a 2600 MHz AMD Opteron machine with 16 GB RAM running Debian Linux. Reported execution time might be overly pessimistic due to a number of other experiments running on this compute server concurrently.

Circuit	# switching aggressors	Switching activity			Time, (CPU)
		LSS	HSS	Random	
s13207	3	9	42	27	2:18:01
s15850	3	10	34	25	1:28:18
s35932	3	25	48	42	20:02:28
s38584	2	19	46	32	6:01:27
s38417	5	12	32	26	2:01:55
Average	3.2	15.0	40.4	30.4	6:22:25

TABLE II

EXPERIMENTAL RESULTS FOR RISING TRANSITION AND $M = N = 10$ CLOCK CYCLES

Larger ISCAS circuits are more challenging in general because there are many more flip-flops than primary inputs and consequently there are many reconvergencies in the unfolded circuit which means more decisions and possibly backtracks. The quality of the found solution is slightly worse than for smaller circuits and the execution time is higher. Both is particularly severe for s35932. Apart from that circuit, the solution quality is largely comparable to that obtained for smaller circuits.

VI. Conclusions

Power droop is a power integrity condition which arises under specific conditions and leads to power starvation and erroneous circuit operation. If power droop is not targeted directly, its effects could be confused with that of random noise, leading to potentially counterproductive mitigation strategies. In contrast, if power droop is identified as such, the parts of the design which need improvements are determined easily.

We proposed an automatic test pattern generation method which creates worst-case power starvation by maximizing the effects of the low-frequency and high-frequency power droop. This necessitates sequential test generation even for circuits with scan. The classical D-algorithm is enhanced by a dynamic constraint generation technique in order to produce a test sequence satisfying non-trivial conditions for power droop maximization. We reported results on ISCAS circuits using a prototype implementation of the algorithm and discussed their quality.

While the proposed implementation is adequate for mid-size blocks and clearly demonstrates the feasibility of the approach in this context, scalability may be limited for larger devices in combination with longer test sequences, i.e., larger number of time frames to unfold. Possible solutions include the use of a basic algorithm more appropriate for sequential test generation such as PODEM in connection with advanced techniques such as static and dynamic learning. Incorporating alternative metrics for power consumption based on accurate delays of the gates is a further possible direction for future research.

VII. Acknowledgment

Parts of this work are supported by the German Research Foundation under grant BE 1176/14-1.

References

[1] C. Tirumurti, S. Kundu, S. Sur-Kolay, and Y. Chang, "A modeling approach for addressing power supply switching noise related failures of integrated circuits," in *Design, Automation and Test in Europe*, 2004.

[2] A. Muhtaroglu, G. Taylor, and T. Rahal-Arabi, "On-die droop detector for analog sensing of power supply noise," *Jour. of Solid-State Circuits*, vol. 39, no. 4, pp. 651–660, 2004.

[3] G. Johnson, "Challenges of high supply currents during VLSI test," in *Int'l Test Conf.*, 2000, pp. 1013–1020.

[4] Y.-S. Chang, S. Gupta, and M. Breuer, "Analysis of ground bounce in deep sub-micron circuits," in *VLSI Test Symp.*, 1997, pp. 110–116.

[5] S. Kundu, "Testing for circuit marginalities," in *Int'l Test Synthesis Workshop*, 2004.

[6] K. Lee, L. Nordquist, and J. Abraham, "Automatic test pattern generation for crosstalk glitches in digital circuits," in *VLSI Test Symp.*, 1998, pp. 34–39.

[7] W. Chen, S. Gupta, and M. Breuer, "Test generation for crosstalk-induced faults: Framework and combinational results," in *Asian Test Symp.*, 2000, pp. 305–310.

[8] Y. Zhao and S. Dey, "Analysis of interconnect crosstalk defect coverage of test sets," in *Int'l Test Conf.*, 2000, pp. 492–501.

[9] R. Kundu and R. D. Blanton, "Identification of crosstalk switch failures in domino cmos circuits," in *Int'l Test Conf.*, 2000, pp. 502–509.

[10] A. Sinha, S. Gupta, and M. Breuer, "Validation and test generation for oscillatory noise in VLSI interconnects," in *Int'l Conf. Computer-Aided Design*, 1999, pp. 289–296.

[11] ——, "Validation and test issues related to noise induced by parasitic inductances of VLSI interconnects," *IEEE Trans. Advanced Packaging*, vol. 25, no. 3, pp. 329–339, 2002.

[12] Y.-S. Chang, S. Gupta, and M. Breuer, "Test generation for ground bounce in internal logic circuitry," in *VLSI Test Symp.*, 1999, pp. 95–104.

[13] ——, "Test generation for maximizing ground bounce for internal circuitry with reconvergent fan-outs," in *VLSI Test Symp.*, 2001, pp. 358–366.

[14] A. Krstic, Y.-M. Jiang, and K.-T. Cheng, "Pattern generation for delay testing and dynamic timing analysis considering power-supply noise effects," *IEEE Trans. on CAD*, vol. 20, no. 3, pp. 416–425, 2001.

[15] Y.-M. Jiang and K.-T. Cheng, "Vector generation for power supply noise estimation and verification of deep submicron designs," *IEEE Trans. VLSI*, vol. 9, no. 2, pp. 329–340, 2001.

[16] M. Nourani, M. Tehranipoor, and N. Ahmed, "Pattern generation and estimation for power-supply noise analysis," in *VLSI Test Symp.*, 2005, pp. 439–444.

[17] M. Nourani and A. Radhakrishnan, "Power-supply noise in SoCs: ATPG, estimation and control," in *Int'l Test Conf.*, 2005.

[18] E. Liau and D. Schmitt-Landsiedel, "Automatic worst case pattern generation using neural networks & genetic algorithm for estimation of switching noise on power supply lines in CMOS circuits," in *European Test Workshop*, 2003, pp. 105–110.

[19] C. Metra, L. Schiano, and M. Favalli, "Concurrent detection of power supply noise," *IEEE Trans. on Reliability*, vol. 52, no. 4, pp. 469–475, 2003.

[20] J. Varquez and J. de Gyvez, "Power supply noise monitor for signal integrity faults," in *Design, Automation and Test in Europe*, 2004, pp. 1406–1407.

[21] J. Wang, X. Lu, W. Qiu, Z. Yue, S. Fancler, W. Shi, and D. Walker, "Static compaction of delay tests considering power supply noise," in *VLSI Test Symp.*, 2005, pp. 235–240.

[22] J. Wang, Z. Yue, X. Lu, W. Qiu, W. Shi, and D. Walker, "A vector-based approach for power supply noise analysis in test compaction," in *Int'l Test Conf.*, 2005.

[23] K. Baker and M. Nourani, "Interconnect test pattern generation algorithm for meeting device and global SSO limits with safe initial vectors," in *Int'l Test Conf.*, 2004.

[24] J. Rearick, B. Eklow, K. Posse, A. Crouch, and B. Bennets, "IJTAG (Internal JTAG): A step toward a DFT standard," in *Int'l Test Conf.*, 2005.

[25] S. Murarka, I. Verner, and R. Gutmann, *Copper—Fundamental Mechanisms for Microelectronis Applications*. John Wiley & Sons, 2000.

[26] Y. Sato, I. Yamazaki, H. Yamanaka, T. Ikeda, and M. Takakura, "A persistent diagnostic technique for unstable defects," in *Int'l Test Conf.*, 2002, pp. 242–249.

[27] M. Pecht, R. Radojic, and G. Rao, *Managing Silicon Chip Reliability*. CRC Press, 1998.

[28] P. Dodd and L. Massengill, "Basic mechanisms and modeling of single-event upset in digital microelectronics," *IEEE Trans. on Nuclear Science*, vol. 50, no. 3, pp. 583–602, 2003.

[29] M. Hsiao, E. Rudnick, and J. Patel, "Peak power estimation of VLSI circuits: New peak power measures," *Trans. on VLSI Systems*, vol. 8, no. 4, pp. 435–439, 2000.

[30] J. Saxena, K. Butler, V. Jayaram, S. Kundu, N. Arvind, P. Sreeprakash, and M. Hachinger, "A case study of IR-drop in structured at-speed testing," in *Int'l Test Conf.*, 2003, pp. 1098–1104.

[31] I. Pomeranz, "On the generation of scan-based test sets with reachable states for testing under functional operation conditions," in *Design Autom. Conf.*, 2004, pp. 928–933.

[32] J. Rearick, "Too much delay fault coverage is a bad thing," in *Int'l Test Conf.*, 2001, pp. 624–633.

[33] X. Liu and M. Hsiao, "Constrained ATPG for broadside transition testing," in *Int'l Symp. on Defect and Fault Tolerance in VLSI Systems*, 2003, pp. 175–184.

[34] Y.-C. Lin, F. Lu, K. Yang, and K.-T. Cheng, "Constraint extraction for pseudo-functional scan-based delay testing," in *Asia and South Pacific Design Autom. Conf.*, 2005, pp. 166–171.

[35] Y.-C. Lin, F. Lu, and K.-T. Cheng, "Pseudo-functional scan-based bist for delay fault," in *VLSI Test Symp.*, 2005, pp. 229–234.

[36] I. Polian and H. Fujiwara, "Functional constraints vs. test compression in scan-based delay testing," in *Design, Automation and Test in Europe*, 2006, in press.

[37] J. Roth, "Diagnosis of automata failures: A calculus and a method," *IBM J. Res. Dev.*, vol. 10, pp. 278–281, 1966.

[38] M. Abramovici, M. Breuer, and A. Friedman, *Digital Systems Testing and Testable Design*. Computer Science Press, 1990.

[39] H. Kriplani, F. Najm, P. Yang, and I. Hajj, "Resolving signal correlations for estimating maximum currents in CMOS combinational circuits," in *Design Automation Conf.*, 1993, pp. 384–388.

[40] C. Wang, K. Roy, and T. Chou, "Maximum power estimation for sequential circuits using a test generation based technique," in *Custom Integrated Circuits Conf.*, 1996, pp. 229–232.

[41] S. Manich and J. Figueras, "Maximizing the weighted switching activity in combinational CMOS circuits under the variable delay model," in *European Design and Test conf.*, 1997, pp. 597–602.

[42] S. Zhao, K. Roy, and C.-K. Koh, "Estimation of inductive and resistive switching noise on power supply network in deep sub-micron cmos circuits," in *Int'l Conf. Computer Design*, 2000, pp. 65–72.

[43] S. Manne, A. Pardo, R. Bahar, G. Hachtel, F. Somenzi, E. Macii, and M. Poncino, "Computing the maximum power cycles of a sequential circuit," in *Desgin Automation Conf.*, 1995, pp. 23–28.

[44] M. Hsiao, E. Rudnick, and J. Patel, "Effects of delay model on peakl power estimation of VLSI sequential circuits," in *Int'l Conf. on CAD*, 1997, pp. 45–51.

[45] Y.-M. Jiang, K.-T. Cheng, and A. Krstic, "Estimation of maximum power and instantaneous current using a genetic algorithm," in *Custom Integrated circuits Conf.*, 1997, pp. 135–138.