Power/Ground Supply Voltage Variation-Aware Delay Test Pattern Generation

Lu Wang, Xutao Wang, Milad Maleki, Bao Liu
The University of Texas at San Antonio
One UTSA Circle, San Antonio, TX, 78249

Abstract—VLSI technology scaling leads to a significant increase in power/ground supply voltage variation and resultant VLSI performance variation, which needs to be taken into account in timing verification and delay test. In this paper, we present a P/G supply voltage variation-aware path delay test pattern generation method. Our experimental results on an AES cipher show that our proposed method finds a maximum of 2.76% power supply voltage drop and 2.62% resultant critical path delay increase, while random test pattern generation finds a maximum of 1.52% power supply voltage drop and 0.31% resultant critical path delay increase in average for two power supply network configurations.

I. INTRODUCTION

As VLSI technology scaling increases parametric variations and resultant performance variability significantly, timing verification and delay test become more and more critical to a successful VLSI design. In the state-of-the-art delay test approaches, power/ground (P/G) supply voltage variation acts an indispensable role. Unlike crosstalk and random dopant fluctuation, P/G supply voltage variation induced gate or interconnect delay variations are more correlated. Without consideration of P/G supply voltage variation, delay test pattern generation may cause failure escape. Moreover, due to higher clock frequency, resultant increased signal toggling rate as well as higher P/G rail parasitic impedance, P/G supply voltage variation becomes increasingly significant to timing verification. Several delay test techniques have been proposed in recent years for maximum supply voltage variation [7] [8] [9], which aim to increase critical path delay and improve delay test performance. However, the maximum supply voltage variation can not guarantee the maximum increased critical path delay and may even result in failure escape which is undesirable in delay test.

Motivated by above observations, instead of relying on the maximum supply voltage variation, we move one step ahead to directly focus on the maximum critical path delay increase. In [9], Todri et al. presented a simulated annealing (SA)-based approach to find patterns that maximize the impact of power supply voltage noise and ground bounce on the critical path delay. But since the existence of reconvergent fanout cycles in the netlist, randomly filling unspecified input patterns may cause logic conflicts and decrease test performance.

To overcome such bottleneck, we present a delay test pattern generation (DTPG) technique which takes into account P/G supply voltage variation effects on timing. Specifically, we first employ vector-based power rail analysis to update supply voltage variation and impacted gate delays. Then we calculate Delta Critical Path Delay ($\Delta D$) for each toggling gate and develop a novel greedy algorithm - Maximum Critical Path Delay (MCPD) - which breaks the reconvergent fanout cycle. At last, we combine traditional ATPG method with MCPD and backtrack to generate the test pattern without randomly assigning.

The rest of this paper is organized as follows. In Section II, we introduce some related works on timing analysis and delay test with consideration of supply voltage variation. Problem formulation is discussed in Section III. Section IV presents overall PGSVV-DTPG flow and our novel Maximum Critical Path Delay algorithm. Experimental results and observations will be shown in Section V before we conclude in Section VI.

II. RELATED WORK

VLSI power/ground supply voltage variation depends on the underlying signal switching activities. In recent years, several methods have been presented to estimate P/G supply voltage noise. Chen and Ling [1] developed a methodology to analyze the power supply noise by integrating the on-chip power bus model and the macro-based switching circuit model with the package-level power distribution model. Zheng et al. [2] built a model which combined all parasitics of the distributed power lines for power noise estimation.

Because P/G supply voltage variation leads to delay deviation and may cause timing violation, timing analysis needs to take this effect into account. Several approaches [3]-[6] are proposed in recent years. Liu et al. [3] conducted timing analysis on possible victim gates to associate voltage drop with a specific extra delay which is weighed by occurrence probabilities of logic 1 or 0 on related nodes. The IR-drop induced weighted extra delay is accumulated for all victim gates in a critical path. Jiang and Cheng [4] first derived the average and standard deviation of all blocks and the correlation coefficients between the blocks, and then estimated the delay. Kim and Walker [5] focused on the spatial correlation of power supply noise and proposed using principal component analysis (PCA) to model power supply noise. The path-delay distribution was then computed with uncorrelated variables. Enami et al. [6] established a statistical power/ground voltage noise model with spatial and temporal correlated power supply noise transformed to uncorrelated variables, and carried out statistical static timing analysis using this model.
Accurate timing analysis forms the basis for path delay test pattern generation. Ma et al. [7] presented a layout-aware delay testing pattern generation procedure under maximum local power supply noise. They selected aggressor cells in close proximity along critical paths and generated switching activities to estimate localized IR-drop, ground bounce and their impact on critical paths, which doesn’t consider global voltage degradation on power and ground supply networks. Krstic et al. [8] proposed a genetic algorithm (GA)-based test pattern generation technique considering power supply noise which aimed to find patterns that sensitizes selected paths and maximizes the power supply noise for the nodes on the target paths. However, the maximum power supply noise may not lead to the maximum critical path delay.

III. PROBLEM STATEMENT

A P/G supply network is a RLC network of distributed parasitic elements - capacitive, resistive and inductive - all of which influence the signal integrity and degrade the performance of the circuit (Fig. 1). Compared with crosstalk induced by capacitance, voltage variation caused by resistance and inductive is the major source for on-chip power supply noise and has more obvious impact on cell delays. Large currents drawn through the resistive power distribution network creates IR-drop $\Delta V$ and sudden changes in current sources cause $L \frac{dI}{dt}$ effects. Both effects slow down the cell transition. In a P/G supply network, each toggling gate contributes as a current source, which can be found in the cell library. The nodal voltage of each gate can be presented as follows:

$$V_{n \times 1} = Z_{n \times m} \cdot I_{m \times 1}$$  \hspace{1cm} (1)

where $V_{n \times 1}$ is the vector of $n$ nodal voltages in the P/G supply network, $I_{m \times 1}$ is the vector of P/G supply currents for each gate, and $Z_{n \times m}$ is the impedance matrix of the P/G supply network.

As previously mentioned, supply voltage variation relies on signal transition activities. Different input test patterns result in different supply voltage variation and timing performance. Thus, the problem of delay test pattern generation with the consideration of P/G supply voltage variation can be formulated as follows:

Problem 1: [Power/Ground Supply Voltage Variation-aware Delay Test Pattern Generation] Given a gate-level netlist $N$, a cell library $L$, a power supply network $P$, and a ground supply network $G$, find the delay test patterns for the top $M$ timing-critical paths considering power/groud supply voltage variation.

IV. POWER/GROUND SUPPLY VOLTAGE VARIATION-AWARE DELAY TEST PATTERN GENERATION

In this section, we first present an overall flow of supply voltage variation-aware delay test pattern generation (SVV-DTPG) in Section IV-A, then propose our novel Maximum Critical Path Delay algorithm in Section IV-B.

A. Overall SVV-DTPG Flow

The overall flow of SVV-DTPG is shown in Fig. 2. We first perform timing analysis with ideal power and ground supply voltages and identify the top $M$ timing critical paths. Then we apply an existing ATPG algorithm to generate the delay test input patterns. Only a portion of primary inputs and internal nets are assigned logic values during this process, and other nets are left as don’t-care value. Due to the existence of reconvergent fanout cycles which will be introduced in Section IV-B, randomly assigning test patterns to primary inputs may cause logic conflict, and cannot guarantee the maximum increased critical path delay, either. We propose a Maximum Critical Path Delay algorithm to break the logic-conflict reconvergent fanout cycles and find the maximum impacted critical path delay in the presence of P/G supply voltage variation. MCPD algorithm applies Delta Path Delay $\Delta D$ on each net in a logic-conflict reconvergent fanout cycle to weigh its impact on critical path delay when it is set a stable logic value. It breaks a reconvergent fanout cycle by setting stable logic to the net with maximum Delta Path Delay $\Delta D$, and generates the complete input patterns without don’t-care values. Details of MCPD algorithm will be presented in Section IV-B.
Due to the chicken-egg dilemma we discussed before, we need an iteration process to check the consistence of input patterns generated by MCPD algorithm. If they are not identical to those in last iteration, we take current generated input patterns for vector-based power rail analysis to update the P/G voltage at each cell. Gate delays are also updated for the next run of timing analysis. We continue this iteration until convergence.

Our approach does not depend on a specific timing analyzer or a specific supply voltage-aware gate delay model. Any timing analysis method with a linear or non-linear delay model is applicable to this proposed SVV-DTPG flow. The differences between our SVV-DTPG flow and the techniques proposed in [8] and [9] which have the similar iteration process are as follows:

1) Our approach applies vector-based power rail analysis for supply voltage variation which is not applied in those two techniques.
2) Instead of assigning input patterns randomly, we employ a MCPD algorithm to update toggling nets and generate convergent test patterns.

Algorithm 1 summarizes this SVV-DTPG technique.

**Algorithm 1: Power/Ground Supply Voltage Variation-aware Delay Test Pattern Generation**

| Input: | a gate-level netlist N, a cell library L, a power supply network P, a ground supply network G |
| Output: | delay test patterns and critical paths |

1) Employ a timing analyzer for signal propagation with ideal power and ground supply voltage networks
2) Apply a traditional ATPG algorithm to assign logic values to primary inputs and internal nets
3) Apply Maximum Critical Path Delay algorithm to generate input patterns
4) Check if new generated input patterns are consistent with those in previous iteration
5) If test patterns are not in consistence
   a) Take test patterns as input for vector-based power rail analysis to get updated voltage value for each gate
   b) Update gate delay with updated supply voltage value
   c) Go back to (1)
6) If test patterns are in consistence, return delay test patterns and critical paths

**B. Maximum Critical Path Delay**

As is mentioned in Section IV-A, there are two folds of difficulties in delay test pattern generation with consideration of P/G supply voltage variation. On one hand, the maximum supply voltage variation may not lead to the maximum critical path delay increase. On the other hand, the presence of reconvergent fanout cycles in the netlist may induce logic conflict at fanout nets. In this section, to overcome these two folds of difficulties, we propose a Maximum Critical Path Delay (MCPD) algorithm to break logic-conflict reconvergent fanout cycles and find test patterns leading to the maximum critical path delay in the presence of P/G supply voltage variation. Before we present the proposed MCPD algorithm, we introduce a few definitions as follows.

**Definition 1:** [Reconvergent Fanout Cycle]

A reconvergent fanout cycle R includes more than one signal propagation path segments which have the same starting point and end point.

Fig. 3 shows a gate-level netlist, where $G - F - C - B - A$ represents a critical path, and nets $i, e, f, b, c, a$ form a reconvergent fanout cycle. Because two paths in this cycle have different parities of logic inversion number, not all the nets can have toggling signals at the same time. Thus, we need to set at least one net stable to resolve the logic conflict. Setting a net stable may imply other nets stable: setting a gate input to a controlling value brings the gate output to the controlled value, while setting a gate output to a non-controlled value implies the gate inputs to the non-controlling value. We introduce *Implied Stable Nets (ISN)* and *Resultant Toggling Nets (RTN)* as follows.

**Definition 2:** [Implied Stable Nets]

The Implied Stable Nets $ISN(i)$ of net $i$ are the set of nets which are implied to be stable when net $i$ is set stable. $ISN(i) = \{ j \mid v(j) \in \{0,1\}, \text{if} v(i) \in \{0,1\} \}$.  

**Definition 3:** [Resultant Toggling Nets]

The Resultant Toggling Nets $RTN(i)$ of net $i$ are the set of nets which can be toggling in the reconvergent fanout cycle when net $i$ is set stable.  

$$RTN(i) = \{ j \mid j \in R \cup j \notin ISN(i), \text{if} v(i) \in \{0,1\} \}.$$  

A toggling net causes supply voltage fluctuation and induces gate delay increase. A rising signal transition leads to power supply voltage drop and delay increase for gates with a rising output signal, while a falling signal transition leads to ground supply voltage bounce and delay increase for gates with a falling output signal. Since the maximum critical path delay may not occur in the presence of maximum supply voltage deviation, we focus on the induced extra delay on a critical path instead of supply voltage variation. We define *Victim Gates* and *Delta Path Delay $\Delta D$* for each net in a logic-conflict reconvergent fanout cycle as follows.

**Definition 4:** [Victim Gates]

Victim gates $S_v(i)$ are the critical path gates at which the P/G voltage is affected by Resultant Toggling Nets $RTN(i)$.  

---

**Fig. 3.** A simple gate-level netlist with a reconvergent fanout cycle. Not all nets can toggle at the same time.

---
while setting net $i$ to a stable logic value $v$ where $v \in \{0, 1\}$.

**Definition 5:** [Delta Path Delay]

Delta path delay $\Delta D_i$ of net $i$ is defined as the maximum delay increase on critical path $C$ due to the supply voltage variation on all the victim gates $m \in S_i$ when net $i$ is changed from a toggling net to a stable net. It is presented as follows:

\[
\Delta D_i = \max \{ \Delta D_i(1), \Delta D_i(0) \}
\]

\[
\Delta D_i(1) = \sum \Delta d(m), m \in S_i(1)
\]

\[
\Delta D_i(0) = \sum \Delta d(m), m \in S_i(0)
\]

where $\Delta D_i(1)$ and $\Delta D_i(0)$ represent critical path delay increase when net $i$ is set to logic 1 and logic 0, respectively.

Due to different toggling signals at the end point of a reconvergent fanout cycle, internal nets have different initialized toggling signals. Each net $i$ has four cases of delta path delay, that is, $\Delta D_i(r, 0)$, $\Delta D_i(r, 1)$, $\Delta D_i(f, 0)$, and $\Delta D_i(f, 1)$, which represent net $i$ is initialized as rising/falling and set to be logic 0/1, respectively. The maximum value is chosen as $\Delta D_i$ which reflects how much net $i$ influences the critical path delay when it is set to be stable.

Before we illustrate the above definitions using the reconvergent fanout cycle in Fig.3, we make several assumptions as follows:

1) With an ideal power/ground supply voltage network, gates 1, 2, 3, 4 in the critical path have delay of 0.045ns, 0.081ns, 0.048ns, 0.030ns, respectively.
2) End point $a$ is initialized as falling signal. Correspondingly, all the internal nets in the reconvergent fanout cycle have toggling signals as shown in Fig.3.
3) Each rising or falling signal transition occurring in the reconvergent fanout cycle induces 1% gate delay increase on victim gates.

Table I gives us Implied Stable Nets (ISN), Resultant Toggling Nets (RTN), Victim Gates (VG) and Delta Path Delay ($\Delta D$) for each net in the cycle. We take net $c$ as an example. When net $c$ is set as logic 0 from falling transition, nets $a, f, g, i, j$ are also implied to change from toggling signal to stable logic. In this case, nets $b, e$ have falling signal transition and each of them increases delay of gate 2, 4 in the critical path by 1%, that is,

\[
\Delta D_c(f, 0) = (\Delta d(2) + \Delta d(4)) \times 1\% \times 2 = (0.081 + 0.030) \times 1\% \times 2
\]

When net $c$ is set as logic 1, there is no implied stable nets. To solve logic conflict at fanout net $i$, we change net $f$ from a rising signal to a falling signal and net $j$ from a falling signal to a rising signal. Then net $i$ is set as a rising signal. Thus, in the cycle, nets $a, b, e, f$ have falling transitions and increase delay of gate 2, 4 by 1%, and net $i$ has rising signal transition and increases delay of gate 1, 3 by 1%, that is,

\[
\Delta D_c(f, 1) = (\Delta d(1) + \Delta d(3)) \times 1\% \times 1 + (\Delta d(2) + \Delta d(4)) \times 1\% \times 4
\]

\[
= (0.045 + 0.048) \times 1\% \times 1 + (0.081 + 0.030) \times 1\% \times 4
\]

\[
= 0.00537
\]

We choose the maximum value between $\Delta D_c(f, 0)$ and $\Delta D_c(f, 1)$ as delta path delay of net $c$, that is,

\[
\Delta D_c = \max \{ \Delta D_c(f, 0), \Delta D_c(f, 1) \} = 0.00537
\]

Similarly, we have delta path delay of net $b, e, f$ in Table I, respectively. Both signal toggling direction and the number of resultant toggling nets decide delta path delay for each net.

We represent a gate-level netlist by a timing graph $G_t = (V, E, I)$, where $V$ is the set of nets, $(u, v) \in E$ if nets $u$ and $v$ are input and output of a gate, and signals across $(u, v)$ are either inverted ($i(u, v) = 1$) or non-inverted ($i(u, v) = 0$) $i(u, v) \in I$. Fig. 4 shows the timing graph $G_t$ of the gate-level netlist in Fig. 3, where solid edges connect two nets with logic inversion $i(u, v) = 1$; dotted edges connect two nets without logic inversion $i(u, v) = 0$; white vertices are nets with rising signal transitions, and black vertices are nets with falling signal transitions.

**Algorithm 2:** Maximum Critical Path Delay

**Input:** A netlist represented by a net connectivity graph $G_t = (V, E, I)$ with nets $V$, gate connections $E$, and logic inversions $I$, a group of critical paths

**Output:** delay test patterns for the critical paths in the presence of P/G supply voltage variation

1) Find all logic-conflict reconvergent fanout cycles $S_{rfc}$.
2) Calculate $\Delta D_i$ for each net $i$ in $S_{rfc}$.
3) Find net $j \in S_i$ with maximum delta path delay $\Delta D_j$ in $S_{rfc}$.
4) Set stable net $j$ and its implied stable nets $ISN(j)$.
5) Set corresponding toggling signals to nets in $RTN(j)$.
6) Update $S_{rfc}$, and go to (3) until $S_{rfc} = \phi$.
7) Set toggling remaining nets in $G_t$.
8) Return input delay test patterns.
Leveraging the Maximum Instantaneous Power algorithm [10] from power analysis, we develop a Maximum Critical Path Delay (MCPD) algorithm to set stable nets with maximum delta path delay which belongs to a logic-conflict reconvergent fanout cycle, but not to a critical path in $G_i$. The algorithm not only breaks logic-conflict reconvergent fanout cycles but also achieves the maximum critical path delay. Algorithm 2 shows the basic steps of MCPD algorithm. We take the timing graph $G_i$ with all the nets $V$, gate connections $E$, logic inversions $I$, and a group of target critical paths as input to generate delay test patterns with consideration of supply voltage variation. We first find all the logic-conflict reconvergent fanout cycles $S_{rfc} \in G_i$, and calculate delta path delay $\Delta D_i$ for each net $i \in S_{rfc}$. Then if there exists logic-conflict reconvergent cycles, find the net $j \in S_j$ with the maximum delta path delay in $S_{rfc}$, and set stable net $j$ and its implied stable nets $ISN (j)$. The nets in $RTN (i)$ are assigned with corresponding toggling signals and $S_j$ is removed out from $S_{rfc}$. Then reconvergent fanout cycles $S_{rfc}$ are updated. We continue this iteration until there exists no reconvergent fanout cycles any more. After all logic-conflict reconvergent fanout cycles are broken, we set toggling signals to the remaining nets in $G_i$, and obtain the delay test patterns which leads to maximum critical path delay.

Taking the timing graph in Fig. 4 as an example, net $c$ has the maximum delta path delay. Following our MCPD algorithm, we assign logic 1 to net $c$ to break this cycle. After that, we can assign falling signal transition to net $f$ and rising signal transition to nets $i,j$. By now, there is no logic conflict in this reconvergent fanout cycle and also maximum critical path delay is kept.

V. EXPERIMENTAL RESULTS AND OBSERVATIONS

Our proposed PGSVV-DTPG algorithm can be applied with any specific timing analysis method. In our experiments, we employ SPSTA [11] as a timing analyzer which provides more signal toggling information. To evaluate the effectiveness of the proposed method, we conduct our experiments on an Advanced Encryption Standard (AES) cipher [12] in different schemes.

We synthesize the gate-level netlist with 3.0ns clock cycle and 45nm Nangate cell library [14]. Table II shows all experiment parameter setup including the number of gate, primary input and output, layer levels of the circuits, clock cycle, temperature, power supply voltage, power pad position, power grid width and input signal information. We assign signal arrival time at the primary inputs and the flip-flop outputs in two different schemes. $P_1$, $P_0$, $P_r$, and $P_f$ represent probability of logic 1, logic 0, rising signal transition, and falling signal transition, respectively. $P_s$ is the signal probability. $\mu_s$ and $\sigma_s$ are mean and variance of signal toggling rate. In the two input assignment schemes, we keep the primary inputs and the flip-flop outputs independent to each other, which means there is zero covariance of signal toggling rate between any two primary inputs, and we assign a mean signal transition temporal occurrence probability in the standard normal distribution $\varphi (t) = N (0, 1)$.

![Fig. 5. PGSVV-DTPG Experiment Flow](image-url)

Our experiment has two steps, as shown in Fig.5. The first step is to prepare all needed files. In this step, we use .ict file which contains the manufacturing parameters from Nangate cell library as input to run Encounter RC Generation and obtain binary technology file(.tch). Combined with circuit physical design and .map file which defines each metal layer and other physical design parameters, the binary technology file(.tch) is applied to Encounter Library Generation to provide power library file which is needed in the next step. The second step is the core of the whole flow. Encounter vector-based power analysis is first run with cell libraries and input patterns(.vcd) from PGSVV-DTPG or random-assigned input patterns(RAND-ATPG) to generate dynamic current file(.ptiavg), which contains detailed current information on each cell. Then Encounter vector-based rail analysis is applied with the dynamic current file(.ptiavg) to produce power supply voltage distribution on each cell(VDD.iv). With VDD.iv and SPEF file, we run ETS VDD-aware timing analysis to generate timing report and SDF file which contains actual timing delay for each cell. Then PGSVV-DTPG method continues to generate input patterns based on the updated SDF file. The whole flow forms an iteration. In the first iteration, PGSVV-DTPG method use ideal gate delay to generate the input patterns, and the whole process stops when the input patterns generated in two successive iteration keep consistent.

we compare our proposed PGSVV-DTPG algorithm with randomly assigning logic to primary inputs(RAND-ATPG) on the worst-case signal arrival time and the average percent of power supply voltage drop. In this experiment, we first com-
We have three observations:

1. For average power supply voltage, random 2-clock-cycle input patterns achieve 0.9823V(1.77% drop) for Power Pad Position I and 0.9873V(1.27% drop) for Power Pad Position II, while 60-clock-cycle input patterns achieve the same percentage of supply voltage drop. In the ideal case without power supply voltage noise, the worst-case signal arrival time is 5.2855ns for Power Pad Position I and 5.6981ns for Power Pad Position II. 2-clock-cycle random input patterns achieve 5.3030ns(0.33% increase) and 5.7152ns(0.30% increase) for two different power pad positions, while 60-clock-cycle random input patterns achieve the same percentage of maximum signal arrival time increase. Input patterns with more clock cycles produce no obvious difference on power supply voltage drop and the worst-case signal arrival time increase. Hence it is sufficient to execute the following experiments between PGSVV-DTPG and RAND-ATPG with 2-clock-cycle input patterns.

2. Observation 2: On the aspect of power supply voltage, our proposed PGSVV-DTPG method achieves 0.9712V(2.88% drop) for Power Pad Position I and 0.9736V(2.64% drop) for Power Pad Position II, while RAND-ATPG method achieves 0.9823V(1.77% drop) and 0.9873V(1.27% drop), respectively. The results show that the input patterns provided by our proposed method induce more supply voltage drop than that of RAND-ATPG.

3. Observation 3: On the aspect of the worst-case signal arrival time, compared with RAND-ATPG, our proposed PGSVV-DTPG technique achieves 5.4233ns(2.61% increase) for Power Pad Position I and 5.8476ns(2.62% increase) for Power Pad Position II on the critical path delay estimation. However, the existing RAND-ATPG method only achieves 5.3030ns(0.33% increase) and 5.7152ns(0.30% increase), respectively. Our proposed approach can provide the input patterns which have more supply voltage variance impact on critical path delay.

VI. Conclusion

Power/ground supply voltage variation has great impact on VLSI performance. In this paper, we proposed Maximum Critical Path Delay algorithm to estimate maximum voltage variation impact on critical path delay and applied it to a novel method of supply voltage variation aware-delay test pattern generation. We compared the worst-case signal arrival time increase and average power supply voltage drop between our proposed approach and an existing randomly assigning input pattern method(RAND-ATPG) on an AES cipher circuit. Experimental results show that our proposed method can achieve 2.76% power supply voltage drop and 2.62% increase on the worst-case signal arrival time estimation for two power pad position schemes on average, while RAND-ATPG method just achieves 1.52% voltage drop on average and 0.31% critical drop.
path delay increment. Our proposed technique can provide the input patterns with more supply voltage variance impact on the critical path delay.

REFERENCES


